

Grado en Ingeniería en Electrónica y Automática Industrial

Trabajo Fin de Grado

# Implementación y evaluación de rotonda de tráfico de vehículos mediante simulación

**Autor:** Fernando Pinedo Peláez

**Tutor/es:** Elisa Sánchez Rojas

2021



UNIVERSIDAD DE ALCALÁ  
*Escuela Politécnica Superior*

**Grado en Ingeniería en Electrónica y Automática Industrial**

*Trabajo Fin de Grado*

**Implementación y evaluación de rotonda de tráfico de  
vehículos mediante simulación**

**Autor:** Fernando Pinedo Peláez

**Tutor:** Elisa Sánchez Rojas

**TRIBUNAL:**

**Presidente:** Miguel Ángel López Carmona

**Vocal 1º:** Isaías Martínez Yelmo

**Vocal 2º:** Elisa Sánchez Rojas

**FECHA:** 27/09/2021



## Contenido

Contenido .....	i
Índice figuras .....	iii
Resumen .....	v
Abstract .....	vii
1. Introducción.....	1
a. Estructura de la memoria .....	2
2. Estado del arte .....	3
2.1 Elementos de las rotondas: .....	3
3.2 Breve resumen histórico sobre las rotondas.....	6
2.3 Herramientas para simulación de tráfico en rotondas.....	17
2.3.1 Traffic Flow de Martin Treiber .....	17
2.3.2 Roundabout Simulator:.....	19
2.3.3 Simulador de Aimsun: .....	20
2.3.4 Sumo:.....	20
2.3.5 Matlab: .....	22
3. Análisis y diseño .....	23
4. Implementación y desarrollo .....	25
4.1 Versión Polar: .....	25
4.2 Análisis de la plataforma de Fuchus .....	29
4.3 Desarrollo del Simulador.....	46
5. Evaluación.....	61
5.1 Rotonda sin semáforos: .....	62
5.2 Rotonda con carriles laterales: .....	62
5.3 Semáforo giratorio: .....	64
5.4 Rotonda con semáforos inteligentes.....	65
5.5 Rotonda con semáforo de 4 sectores: .....	66
5.6 Semaforo circular con dos sectores intercambiados:.....	68
5.7 Semáforo circular con tres colores cambiados. Versión 1: .....	69
5.8 Semáforo circular con tres colores cambiados. Versión 2: .....	70
5.9 Comparación: .....	71
6. Conclusiones y Trabajo futuro.....	75
7. Referencias: .....	79



## Índice figuras

Figura 1. Rotonda de Marasti en Rumanía, mayo de 1998 [0] .....	1
Figura 2. Posibles puntos de colisión. Cruce vs rotonda .....	3
Figura 3. Rotondas por trenzado .....	7
Figura 4. Prioridad a las entradas o a la derecha .....	8
Figura 5. Prioridad en el anillo o a la izquierda .....	9
Figura 6. Rotonda partida .....	10
Figura 7. Intersección demasiado grande .....	11
Figura 8. Rotonda doble .....	11
Figura 9. Turbo-rotonda .....	12
Figura 10. Turbo-rotonda frente a rotonda clásica. ....	13
Figura 11. Magic Roundabout, Swindon[13] .....	13
Figura 12. Ejemplo de rotonda mal planificada. ....	15
Figura 13. Rotonda innecesaria 1 .....	16
Figura 14. Rotonda innecesaria 2 .....	16
Figura 15. Simulador .....	17
Figura 16. Prioridad al anillo 70 segundos .....	18
Figura 17. Rotonda bloqueada .....	18
Figura 18. Roundabout Simulator .....	19
Figura 19. Sistema de control de tráfico mediante simulación. Aimsun. ....	20
Figura 20. Plano general de una simulación en SUMO .....	21
Figura 21. SUMO, ¿Qué puede simular?[17] .....	21
Figura 22. Intersección en SUMO (izquierda) Motocicletas (derecha) .....	22
Figura 23. Boceto de la rotonda .....	23
Figura 24. Incorporaciones y salidas .....	24
Figura 25. Concepto de la rotonda .....	25
Figura 26. Detalle salida este .....	26
Figura 27. Matriz G .....	26
Figura 28. Rotonda con vehículos y representación de colores y formas. ....	29
Figura 29. Representación cartesiana .....	30
Figura 30. Muestra matriz cases(1:15, :) .....	31
Figura 31. Muestra matriz cars(1:38,:) .....	34
Figura 32. cars 1 coche .....	38
Figura 33. Output simulación 1 coche .....	38
Figura 34. Trayectoria del coche a través de la rotonda .....	39
Figura 35. Matriz cars. 35 coches. ....	40
Figura 36. t=15, 35 coches .....	41
Figura 37. t=22. 35 coches .....	41
Figura 38. t=25. coches en espera .....	42
Figura 39. Vaciado de la rotonda .....	43
Figura 40. Estado de la cola de coches. ....	44
Figura 41. Rotonda saturada, carga elevada .....	45
Figura 42. Resultado final .....	45
Figura 43. Interfaz .....	47
Figura 44. Rotonda con carriles laterales .....	48
Figura 45. Listbox selección parámetros .....	48
Figura 46. Concepto semáforo giratorio. ....	49
Figura 47. Semáforos inteligentes .....	50
Figura 48. Semáforo 4 sectores .....	50
Figura 49. Semáforo 4 sectores dinámicos .....	51

Figura 50. Gráficos de barras.....	54
Figura 51. Configuración flujo vehículos .....	61
Figura 52. Configuración. ....	61
Figura 53. Rotonda estándar.....	62
Figura 54. Configuración .....	63
Figura 55. Rotonda con carriles laterales. ....	63
Figura 56. Saturación de la rotonda. ....	64
Figura 57. Semáforo circular .....	64
Figura 58. Semáforos inteligentes y configuración.....	65
Figura 59. Tiempo medio. 100 coches izquierda 185 coches derecha.....	66
Figura 60. Semáforo 4 sectores. Configuración inicial .....	67
Figura 61. Semáforo circular 185 coches .....	67
Figura 62. dos colores cambiados.....	68
Figura 63. SC con sectores azul magenta intercambiados 100 y 185 coches 45s .....	68
Figura 64. tres colores cambiados .....	69
Figura 65. Izquierda 100 coches/45s. Derecha 185 coches/ 45s.....	69
Figura 66. Tres colores cambiados. ....	70
Figura 67. Izquierda 100 coches/45s. Derecha 185 coches/ 45s.....	70
Figura 68. Comparación .....	71
Figura 69. Resultado comparación 100 coches $t_{max}=45$ .....	71
Figura 70. 200 coches $t_{max}=45$ .....	72
Figura 71. Resultado 185 coches.....	72
Figura 72. Rotonda con carriles laterales con semáforo inteligente. 50 iteraciones .....	73
Figura 73. Rotonda con carriles laterales sin semáforos. 50 iteraciones .....	74
Figura 74. Parámetros semáforo giratorio .....	76



## Resumen

Las rotondas buscan solucionar los problemas de saturación del tráfico en las vías urbanas. Vamos a hablar de los elementos que forman una rotonda, cómo surgieron las primeras rotondas y como han evolucionado. El objetivo es, mediante el uso de un simulador, buscar posibles mejoras que ayuden a agilizar el tráfico de las rotondas incluso en las peores condiciones. Para ello hemos realizado un simulador de una rotonda en Matlab, con el que se ha visto cómo afectaría a una rotonda la aplicación de diferentes modificaciones.

Palabras clave: Matlab; Rotondas; Simulador;



## Abstract

Roundabouts try to solve the saturation problem on urban roads. We will talk about the elements that form a roundabout, how the first roundabouts appeared and how they evolved. The objective is, through the use of a simulator, to look for possible improvements that help to speed up the traffic of the roundabouts even in the worst conditions. For this we made a simulator of a roundabout in Matlab, we checked how would affect to carry out different modifications in a roundabout.

Keywords: matlab; roundabout; simulator;



## 1. Introducción

Cuando nos centramos en el tráfico vehicular, las ventajas que ofrecen las rotondas frente a intersecciones estándar han provocado que su uso se extienda cada vez más por el mundo durante los últimos años, si bien son particularmente populares en Europa. Su gran versatilidad las convierte en una solución aplicable prácticamente en cualquier situación. Sin embargo, no todos los escenarios de tráfico se ven favorecidos por su uso. Por ejemplo, cuando el tráfico no está bien repartido la circulación puede verse afectada de forma negativa. Una gran carga de tráfico en una de las vías de acceso puede dar lugar a aglomeraciones de vehículos o, en otras palabras, atascos. Este tipo de escenarios se pueden dar con relativa frecuencia cuando una rotonda da acceso a una autopista, es lógico pensar que gran parte de los vehículos se dirigirán en esa dirección.



*Figura 1. Rotonda de Marasti en Rumanía, mayo de 1998 [0]*

En el presente Trabajo Fin de Grado (TFG) se plantea, como principal objetivo, estudiar configuraciones alternativas (no estándar) para una rotonda. Para realizar el mismo, se plantea el uso de simuladores. Un simulador en este caso será un programa que representa el funcionamiento de un fenómeno, una rotonda, y las condiciones a la que esta está sometida. Más concretamente, en este TFG, se va a buscar una forma de representar tanto la rotonda como los coches que circulan por ella en un entorno gráfico, que será Matlab. Primero buscaremos crear un esquema básico de la rotonda. En ella los coches tendrán un destino predefinido y unos movimientos limitados en función de su origen y de su destino. Después se ampliarán las opciones del simulador, añadiendo la parametrización del tráfico, semáforos y añadiendo distintos tipos de rotondas.

En resumen, para alcanzar el objetivo principal del TFG, se plantean los siguientes pasos:

- ) **Revisión estado del arte:** Se revisaron varios trabajos similares a este, encontrando el trabajo de Fuchus, del que hablaremos en el capítulo 4.
- ) **Análisis de Matlab:** consideración de las distintas herramientas que ofrece Matlab que puedan ayudarnos en la realización de nuestro simulador.
- ) **Interfaz Gráfica:** diseño de una interfaz con la que representar la rotonda en nuestro simulador.
- ) **Análisis Fuchus:** Estudio del trabajo y de cómo podemos utilizarlo para realizar nuestro simulador.
- ) **Desarrollo simulador:** Preparación de las herramientas que nos ayudarán a buscar una solución que agilice el tráfico de una rotonda.
- ) **Evaluación:** Análisis de los resultados obtenidos con nuestro simulador.

#### a. Estructura de la memoria

La presente memoria consta de 7 capítulos:

En el capítulo 1 es esta introducción.

Capítulo 2 Estado del arte. Veremos un resumen histórico de las rotondas. Hablaremos de sus componentes y de distintos simuladores.

Capítulo 3 Análisis y diseño. Se habla del trabajo previo que existía antes de este trabajo y de cómo lo utilizaremos en nuestro proyecto.

Capítulo 4 Implementación y desarrollo. Se detallan los distintos pasos que se ha dado en el desarrollo del trabajo.

Capítulo 4 Evaluación. Veremos los resultados obtenidos.

Capítulo 5 Conclusiones y futuro trabajo. Veremos si el simulador ha cumplido los objetivos y algunas funcionalidades que no se ha llegado a implementar.

Capítulo 7. Referencias

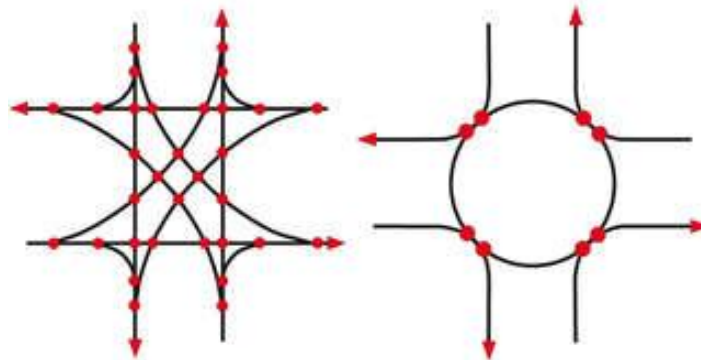
## 2. Estado del arte

En el presente Capítulo, primero realizaremos un breve paseo cronológico sobre la evolución de las intersecciones y la aparición de las primeras rotondas. A continuación, vamos a ver los componentes fundamentales de las rotondas.

### 2.1 Elementos de las rotondas:

Las rotondas, también llamadas en distintas partes del mundo glorietas, óvalos, redomas o redondeles [1], son un cruce especial que se caracteriza por la manera en la que se circula en los tramos que unen las vías que confluyen en ella. Se circula a través de un anillo que conecta todas vías dando lugar a una isleta central. El cambio de trayectoria de los vehículos hace que no se crucen de forma secante, como podría pasar en una intersección habitual, sino que se cruzarían de forma tangencial, reduciendo los puntos de conflicto como se ve en la Figura 2.

En la siguiente imagen de [2], podemos comparar los posibles puntos de accidente en una intersección normal frente a los que encontraríamos en una rotonda. En una intersección de 2 calles, cada una de ellas con dos carriles, uno para cada sentido de la circulación, se tendrían 32 posibles puntos en los que se podrían producir colisiones.



*Figura 2. Posibles puntos de colisión. Cruce vs rotonda*

En la Figura 2, podemos ver el significativo incremento en la seguridad vial que supone una rotonda, frente a la alternativa de un cruce simple de dos calles. Además de las ventajas en seguridad, la rotonda supone una mejora en la fluidez del tráfico por la intersección, ya que el flujo por la rotonda es constante, salvo en caso de la saturación de esta, de lo que se hablará más adelante.

La clave del funcionamiento de las rotondas reside en la prioridad que se adjudica a los vehículos que circulan dentro del anillo, por encima de aquellos que quieren entrar. Independientemente de la maniobra a realizar, un vehículo que se aproxima a la glorieta ya quiera seguir de frente, girar a la derecha, izquierda, o realizar un cambio de sentido, lo único que debe hacer es esperar a que haya un intervalo entre los vehículos que circulan por el interior del anillo para incorporarse con seguridad a la calzada. Una vez se han incorporado a la calzada anular, deben seguir su sentido de circulación (en España sentido

contrario a las agujas del reloj) hasta alcanzar la salida que se corresponda con su destino. Dado que los vehículos que circulan por dentro de la rotonda tienen prioridad, no hay que preocuparse por encontrarse con otro vehículo en mitad del trayecto (en principio), lo que facilita la circulación y aumenta la seguridad.[3]

Calzada anular: Tramo principal de la rotonda por donde circulan los vehículos. Llamada de esta forma debido a la forma, habitualmente circular, que caracteriza las rotondas.

Islote central: La zona destinada a la circulación de vehículos discurre alrededor del islote central, una zona que funciona como obstáculo para los vehículos que se aproximan a la rotonda, obligándoles a reducir la velocidad y a cambiar de dirección. En algunas mini glorietas, como las diseñadas por Blackmore Frank, el islote central es transitable para aquellos vehículos que, por su longitud, no pueden maniobrar por el carril anular de forma segura. Esto es habitual en algunos países europeos, no siendo este el caso de España. Se trata de una solución útil especialmente cuando se quiere transformar una intersección habitual en una rotonda, y no se disponga de espacio suficiente para construir una rotonda lo suficientemente grande.

Entradas y salidas: Son los puntos donde las vías que participan en la intersección convergen. Para facilitar la incorporación de los vehículos, o su salida, se curva la entrada o salida ligeramente. Dadas las reglas a las que están sujetos los vehículos durante la circulación por las rotondas, las entradas deberían tener el mismo número de carriles que la calzada central. Si una entrada tiene más carriles que de los que se dispone para circular en el interior de la rotonda puede darse lugar a colas, aun cuando no haya un tráfico elevado. Para las salidas la estrategia de diseño es distinta, se busca un radio de la curva mayor para facilitar la salida de los vehículos.

La clave del funcionamiento de las rotondas hoy en día es que la prioridad de paso la tienen los vehículos que circulan por el anillo. Por esto, la señalización de las maniobras es clave a la hora de proporcionar información a los demás usuarios. Así la circulación por la rotonda mejora al proporcionar información acerca de la trayectoria del vehículo, permitiendo a los usuarios que están situados en una entrada esperando para poder acceder al anillo central, actuar con mayor rapidez mejorando el funcionamiento de la rotonda. Esto no solo mejora el flujo de vehículos, sino que aumenta la seguridad de los usuarios, tanto de los vehículos como de peatones que se encuentren en las inmediaciones.

En rotondas con más de un carril en el anillo interno, es fundamental la correcta utilización de los carriles, tanto de los carriles internos como los de aproximación a la rotonda, ya que esto afectará tanto a la seguridad de los usuarios, como a la fluidez del tráfico. La incorrecta elección de los carriles puede dar lugar a cruces innecesarios en las trayectorias de vehículos, aumentando la posibilidad de un accidente, provocando detenciones o la reducción de velocidad de algún vehículo, que en otro caso se habría evitado.

Por ejemplo, supongamos una rotonda con dos carriles en el anillo interno y que las vías de entrada a la rotonda también tienen dos carriles cada una, lo que sería lo ideal. Además, en este supuesto en la rotonda convergen cuatro calles,



dando lugar a cuatro entradas y sus correspondientes cuatro salidas. En esta situación un vehículo que se aproxime a la rotonda tendrá cuatro posibles opciones dependiendo de la salida que tenga intención de tomar:

- Tomar la primera salida: El vehículo debería acercarse a la rotonda por el carril derecho de la vía de entrada y circular por el carril derecho del anillo hasta llegar a su salida. La única precaución que necesita tomar el conductor es asegurarse de que, en el momento de entrar a la calzada circular, dispone de tiempo suficiente para realizar la maniobra sin poner en peligro su seguridad o la de los demás. Si hubiera un segundo vehículo, que se aproxime a la rotonda por la misma vía que el primero, pero por el carril izquierdo. El conductor del primer vehículo puede deducir por el comportamiento del segundo, que este no tiene intención de tomar la primera salida, por lo tanto, tampoco utilizará el carril exterior del anillo, pudiendo el primero circular con seguridad y sin ralentizar su marcha ni la de usuarios que circulen por detrás de él. De forma que la trayectoria de estos vehículos no se cruzará en ningún momento durante el tiempo que utilicen la rotonda.
- Tomar la segunda salida: El proceso será idéntico al caso anterior. Continuará circulando por el carril derecho hasta que llegue a la segunda salida y pueda salir por ella.
- Tomar la tercera salida: Un vehículo aproximándose a la rotonda, con intención de salir por la tercera salida, debería hacerlo por el carril izquierdo para acceder también al carril izquierdo del anillo. Circulará por el carril interior, pasando del carril interior al exterior cuando la siguiente salida que encontremos en la rotonda sea por la que vamos a circular.
- Realizar un cambio de sentido, cuarta salida: De nuevo, se usará el carril izquierdo para tomar el carril interno de la rotonda. Se circularía por el carril interno hasta haber superado la salida anterior a la que vamos a usar. Momento en el que se cambiará al carril externo para salir de la rotonda por la cuarta salida.

En todos los casos, dado que son los vehículos que circulan por la calzada anular los que tienen prioridad, frente a los que se aproximan a la rotonda, no se encontrarán con otro vehículo que se cruce en su trayectoria.

Como todo proceso sujeto a las decisiones humanas, puede haber y habrá errores. Un usuario podría equivocarse a la hora de elegir el carril por el que debería circular para llegar a su salida, calcular mal el tiempo del que se dispone para entrar a la rotonda, adelantar o retrasar demasiado la maniobra de entrada en la rotonda, o incluso obligar a un vehículo o varios a frenar, o peor, a modificar su trayectoria, lo que implicaría a otros vehículos. Cualquier error disminuirá la efectividad en el funcionamiento de la rotonda, dando lugar a retrasos y/o a la acumulación de vehículos, que en otra situación podrían circular de forma fluida. Estos errores, aunque pueden ser previstos en ciertas ocasiones, de manera general son muy difíciles de predecir y analizar. Por ello, hay que adaptarse a las distintas situaciones que puedan surgir.

Por ejemplo, si un conductor se equivoca eligiendo el carril para circular antes de tomar nuestra salida, será mejor dar una vuelta a la rotonda y que se aproxime por el carril adecuado a la salida, en lugar de arriesgarse a cruzar desde el carril interno de la rotonda hasta el carril exterior para coger la salida, provocando una posible colisión con el vehículo que circule por el carril exterior del anillo. Otro comportamiento incorrecto en la circulación por la rotonda podría ser la espera para acceder a la rotonda con parte del vehículo en el interior del anillo. Esto puede estar motivado tanto por una incorrecta construcción de la rotonda, dando lugar a una falta de visibilidad a la hora de acceder a la rotonda, como por la personalidad del conductor. En este sentido, ayuda a evitar este comportamiento la correcta visibilidad desde la línea de ceda el paso.

Según un estudio en 12 glorietas de la Comunidad de Madrid [4]. La entrada de un vehículo al anillo, mientras otro vehículo circula por este, es una maniobra que aumenta la peligrosidad de la intersección a la vez que no aumenta de forma significativa la capacidad de la rotonda, ya que la realizan uno de cada mil vehículos. Aunque estas no deberían afectar a la seguridad, ya que los vehículos implicados circularán hacia salidas distintas, y por lo tanto sus trayectorias no se cruzarán. Si se da que la línea de ceda el paso es demasiado ancha para adecuarla al número de carriles de la rotonda, podrían producirse situaciones en las que el número de coches que pretenden entrar a la rotonda sea mayor que el número de carriles dentro de esta, dando lugar a acelerones y frenazos. Esto es más frecuente en rotondas congestionadas.

### 3.2 Breve resumen histórico sobre las rotondas

Antes de la aparición de las primeras rotondas, ya existían cruces circulares como es el caso de la plaza de l'Étoile o plaza de la estrella, ahora llamada plaza Charles de Gaulle, en la que se encuentra el arco del triunfo, gran plaza en la cual convergen 12 avenidas [5].

Fue el arquitecto francés Eugène Hénard (1849-1923) quién proyectó en París las primeras glorietas urbanas.

Otra intersección digna de mención es Columbus Circle, en Manhattan, terminado en 1905 y creado por el empresario y pionero en seguridad vial William Phelps Eno y, aunque esté sujeto a debate, es considerado el primer anillo vial. Columbus Circle fue nombrada la mejor rotonda por la *roundabout appreciation society* [6], un grupo del Reino Unido dedicado al debate sobre rotondas.

En 1907 se instauró la circulación giratoria en un único sentido en torno al Arco del Triunfo, en la plaza anteriormente mencionada Charles de Gaulle.

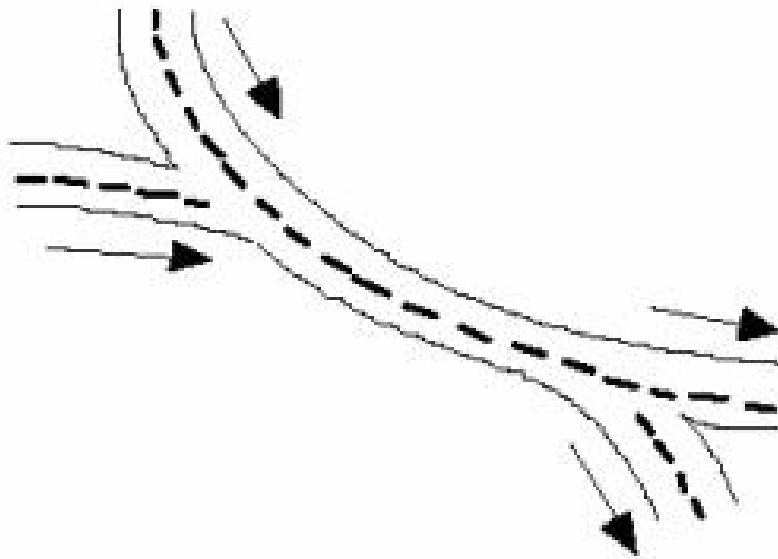
Aunque la que se considera la primera glorieta fue construida en Letchworth en Inglaterra en 1909, originalmente diseñada como una isla de tráfico para los peatones.

En Londres no fue hasta 1925 cuando se construyó la primera rotonda situada en el centro de la ciudad, en aquel entonces eran los vehículos que circulan fuera de la rotonda los que tenían prioridad. Hasta 1966, las rotondas en Inglaterra

funcionan según la norma de prioridad a la izquierda, en el resto de los países a la derecha. Los vehículos entrantes circulan a más velocidad que los que circulan por el anillo, forzando la entrada y obligando a reducir aún más la velocidad de los que circulan por dentro. Esto favorece la entrada de los vehículos que provienen de alguna rama, obligando incluso a que los que ya están en ella se detengan, lo que provocaba una cola en la calzada anular que imposibilitaba la entrada o salida, bloqueando cualquier movimiento.[7]

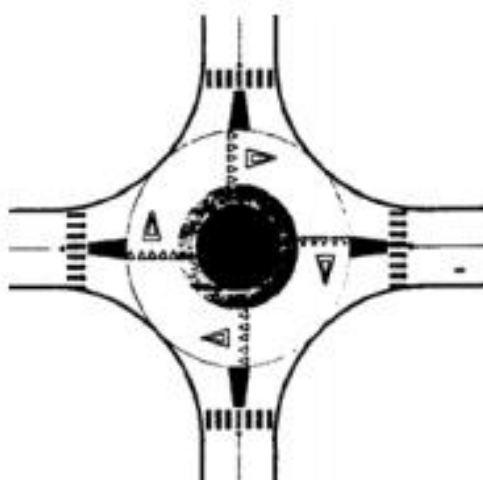
La solución al problema del autobloqueo llega en la década de 1960, cuando los ingenieros ingleses experimentan con la inversión de las prioridades. La prioridad para los vehículos que se sitúan dentro de la glorieta se establece en Inglaterra en 1966, en Francia en 1984, en Suiza en 1987 y en España en 1990 [8]. Este cambio permitió reducir la distancia entre entradas y salidas consecutivas, reduciendo de forma significativa el diámetro, lo que permite una generalización de la implantación de las rotondas.

Las rotondas modernas tal y como las conocemos hoy día, no nacen hasta la década de 1960 de mano de ingenieros como Blackmore Frank, quién lideró el desarrollo de las reglas que regulan el tráfico en las rotondas. En España la primera glorieta se construyó en Palmanova, en Mallorca, en 1976.



*Figura 3. Rotondas por trenzado*

A mediados del siglo XX se proyectaban en Europa las primeras rotondas, que funcionaban por trenzado. Estas rotondas alcanzaban tamaños considerables debido a su funcionamiento. En ellas no existía regla de prioridad alguna, por lo tanto, los vehículos que circulaban por la rotonda se mezclan con los que entran, produciendo el efecto de trenzado. Si la longitud no era lo suficientemente grande podía dar problemas de autobloqueo.

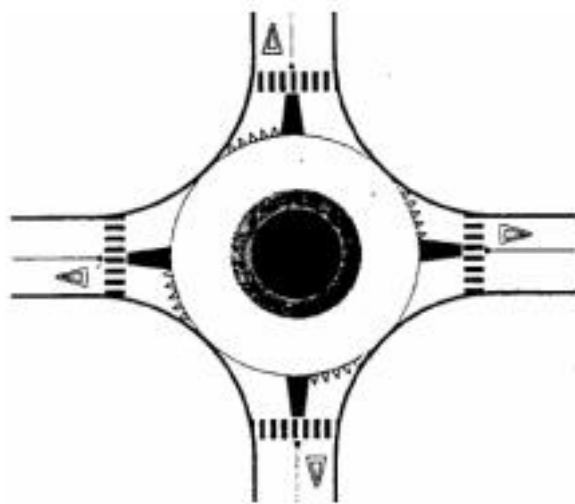


*Figura 4. Prioridad a las entradas o a la derecha*

Después aparecen las rotondas con prioridad a la derecha (Figura 4) , o prioridad a las entradas. Se establece que son los vehículos que se aproximan a la rotonda los que tienen la prioridad de paso frente a los que ya circulan por el anillo. En este tipo de rotondas el fenómeno de autobloqueo está muy presente, por lo que tienden a aumentar su tamaño. En estas rotondas, son los vehículos que circulan por dentro del anillo los encargados de reducir su velocidad para ceder el paso a los vehículos que se acerquen a la rotonda.

Esto supondrá que la velocidad media de los vehículos en el anillo se reduce considerablemente, por lo que el flujo de vehículos a través de la rotonda se verá ralentizado. Además, incrementa la peligrosidad de la intersección, los vehículos que circulan por dentro del anillo deberán mantenerse alerta ante un posible vehículo que se aproxime a la rotonda y también vigilar que el vehículo que circula delante de ellos en el anillo no reduzca la velocidad demasiado. Todo esto da lugar a una rotonda ineficaz salvo quizá en condiciones determinadas de tráfico, por ejemplo, una intersección, por la que la mayor parte de los vehículos lleguen desde una vía y salgan por la primera salida. Mientras que por las demás vías que converjan en la rotonda presenten únicamente tráfico puntual, que llegaría a la rotonda y podría pasar solo si el tráfico por la vía más transitada lo permite, teniendo que esperar mientras no sea así.

Para evitar el autobloqueo de las rotondas se modifica la prioridad, para que sean los vehículos que circulan por dentro del anillo los que tengan la prioridad de paso. La diferencia está en dónde se sitúa la línea de ceda el paso, en estas rotondas dicha línea estará situada donde se junte la vía de acceso con la rotonda, a diferencia de las rotondas con prioridad a la derecha, que situaban la línea de ceda el paso donde la calzada anular interseca con el carril de acceso a la rotonda.

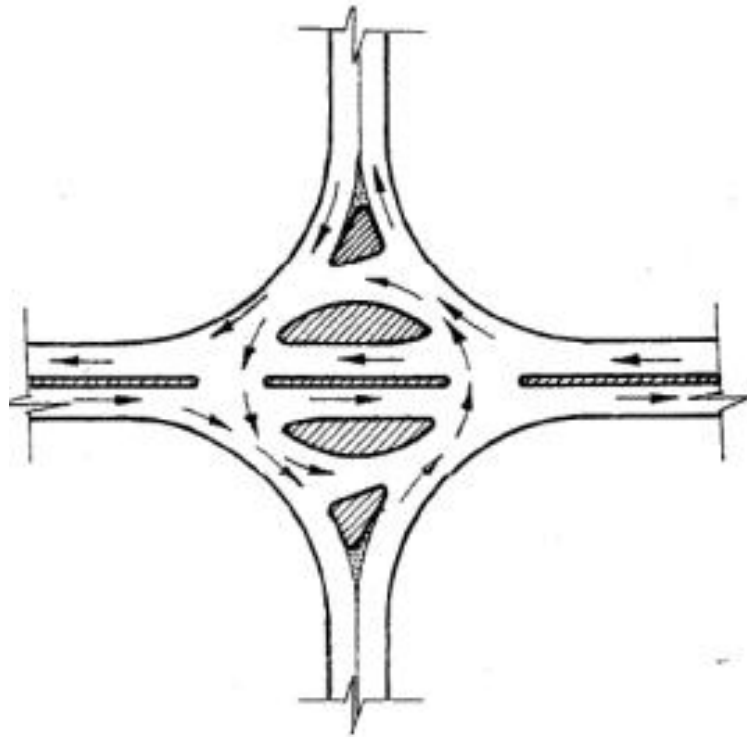


*Figura 5. Prioridad en el anillo o a la izquierda*

Aquí son los vehículos que se aproximan a la rotonda los que deben reducir la velocidad, permitiendo que los que circulan por el anillo puedan hacerlo de forma segura y fluida. Como en ningún caso el tráfico dentro del anillo debe verse interrumpido, esto hará que la velocidad de los vehículos dentro de la rotonda sea mayor, mejorando el flujo, reduciendo los autobloqueos y la congestión de las vías de acceso a la glorieta. Como los vehículos que circulen por dentro del anillo tendrán prioridad en todo caso frente a los que intenten acceder a la rotonda, únicamente deben concentrarse en llegar a salvo a su salida y abandonar la rotonda. Serán los vehículos que se aproximen a la rotonda los que deban vigilar que puedan acceder al anillo de forma segura, reduciendo la velocidad y deteniéndose si es necesario. Por lo tanto, las posibilidades de que haya un accidente serán más bajas que en una rotonda con prioridad a la derecha.

Como a pesar de la mejora que supone la prioridad en el anillo frente a la prioridad a las entradas, puede darse que, por las características del tráfico de una zona, una rotonda no sea capaz de autorregular su tráfico y se produzcan grandes retenciones. Se pueden instalar semáforos en algunas rotondas con el objetivo de regular su tráfico. Por lo general esto suele darse en rotondas urbanas, donde la gran carga de tráfico dificulta su autorregulación. En una zona urbana, una rotonda con una carga de tráfico demasiado alta puede dar lugar a colas y retenciones que se propaguen y afecten a otras calles más alejadas de la rotonda, afectando al tráfico de toda la zona. Además, en zonas urbanas suele haber tráfico peatonal, por lo que puede ser necesaria la utilización de semáforos para regular el paso, tanto de peatones como de vehículos.

Lo más habitual es que las rotondas tengan una planta circular. El islote central es un círculo y los carriles anulares son circulares. Esto hace que la rotonda sea más intuitiva, más simple y fácil de comprender por los usuarios, lo que reducirá su peligrosidad mejorando y favoreciendo el flujo de vehículos a través de ella. La curva constante hace que se pueda circular a través de ella sin necesidad de mover el volante.



*Figura 6. Rotonda partida*

Si por alguna razón no se puede ejecutar la rotonda de forma circular, también se puede hacer elíptica, la excentricidad recomendada está entre  $\frac{3}{4}$  y 1 [9].

A la rotonda se le pueden añadir carriles atravesando el islote central, de forma que las vías que quedan en lados opuestos quedan conectadas. Esto sería una glorieta partida, según la ley de tráfico:

“En las denominadas glorietas partidas, dos tramos—generalmente opuestos—se conectan directamente a través de la isleta central, por lo que el tráfico que pasa de uno a otro no la rodea” [10] Figura 6.

Este tipo de glorieta, en caso de no disponer de señalización, se aplicarán las preferencias de paso de una intersección normal. Teniendo la preferencia aquellos vehículos que se aproximen a la glorieta. En la Figura 6 podemos distinguir un tramo principal (horizontal) que atraviesa la isleta central, y un tramo secundario (vertical) que rodearía la isleta. El tramo principal que atraviesa la rotonda, el horizontal en este caso estará reservado a los vehículos que saldrán de la rotonda a través de la vía principal, por la derecha o por la izquierda. La calzada anular se usará tanto para vehículos que circulen por el tramo principal y que deseen hacer un cambio de sentido, como para los vehículos que lleguen desde el tramo secundario.

Para cruzar la glorieta se seguirá recto en el caso del tramo principal, y se circulará por el anillo en caso del tramo secundario, cediendo el paso a los vehículos que se aproximen por la derecha. Los giros a la derecha se realizarán accediendo a la calzada anular y saliendo por la primera salida. En caso de querer cambiar de sentido, en el tramo secundario se rodeará la glorieta, y en el tramo principal habrá dos posibilidades. Bien se puede rodear parcialmente la



glorieta y acceder al carril que atraviesa el islote central, o rodear completamente el islote.

Para un giro a la izquierda desde el tramo principal se bordeará la calzada anular como si de una glorieta común se tratara, teniendo en cuenta que no se tiene prioridad al circular por la calzada circular. Desde el tramo secundario se bordeará parcialmente la rotonda pudiendo acceder al tramo principal a través del carril que atraviesa el islote, o rodear completamente este.

Si el área en el que confluyen las vías es demasiado extensa, o si hay algún obstáculo en la zona, como una vía de ferrocarril, un río, otra carretera, se puede utilizar una rotonda doble para resolver la intersección.

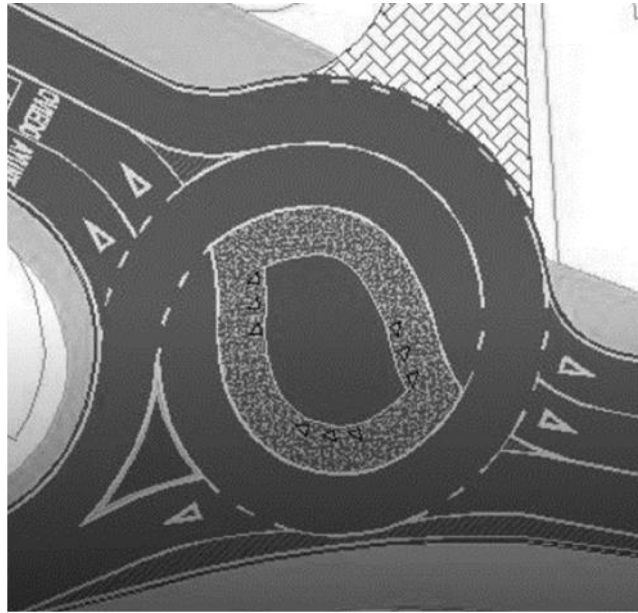
Una rotonda doble resulta de la adaptación de una rotonda demasiado grande, que se ha transformado en un sistema de dos rotondas que están unidas por un tramo común. En la Figura 7 se puede ver una intersección demasiado grande para una sola rotonda y en la Figura 8 una rotonda doble donde el tramo común cruza por encima de otra carretera.



*Figura 7. Intersección demasiado grande*



*Figura 8. Rotonda doble*



*Figura 9. Turbo-rotonda*

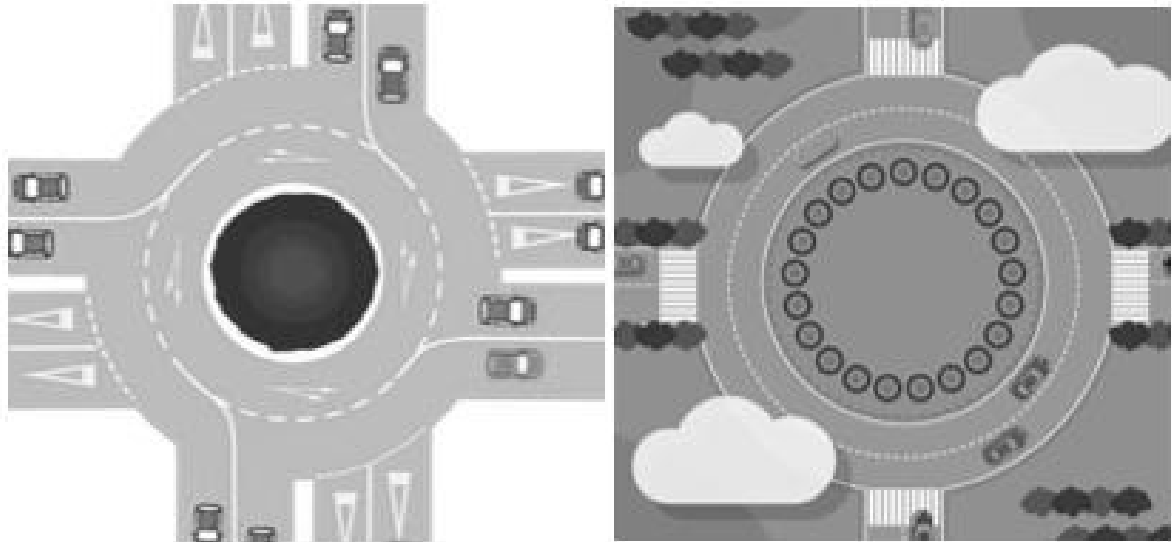
Una posible evolución de las rotondas son las llamadas turbo-rotondas o rotondas inteligentes. En ellas los carriles están delimitados por líneas continuas, de forma que los vehículos que circulen por la rotonda no puedan cambiar de carril mientras se dirigen a su destino. Cada carril dentro de la calzada anular llega a una salida, de esta forma se evitan cambios de carriles. Estas aparecen a finales de los años 90 en Holanda[11] con el objetivo de agilizar el tráfico en rotondas de pequeñas dimensiones en zonas con una carga de tráfico elevada, lo que provoca la saturación de la rotonda.

“Según datos de la DGT, un 45% de los conductores usa el carril interior para salir de la rotonda y un 15% traza las rotondas rectas” [11]

Según un estudio de Formaster[12] “El 80% de los conductores comete un error al circular por una rotonda”

Las marcas viales que caracterizan las rotondas inteligentes facilitan su uso. Ya que no hace falta tener claras las normas relativas a la circulación a través de una rotonda. Si vas por el carril exterior tendrás que salir de la rotonda cuando las marcas viales así lo indiquen. Esta ausencia de cambios de carril innecesarios reduce la siniestralidad de la rotonda. Siendo esto especialmente notable para los usuarios más vulnerables, como los motoristas, para quienes cualquier colisión con un vehículo despistado que cambia de carril cuando no debería podría ser fatal.





*Figura 10. Turbo-rotonda frente a rotonda clásica.*

Hay que destacar la rotonda mágica de Swindon, Reino Unido, Figura 11 . Existe desde hace 60 años.



*Figura 11. Magic Roundabout, Swindon[13]*

Está constituida de una rotonda en el centro, a la que llamaremos principal, en la que se circula en sentido antihorario. Alrededor de esta hay otras 5 rotondas, las llamaremos satélites, en las que los vehículos giran en sentido horario. Uniendo todas las rotondas exteriores, formando la última rotonda, está la séptima rotonda, a la que llamaremos rotonda exterior. Una calzada anular que circula alrededor del conjunto de todas las rotondas ya mencionadas.

Esta rotonda mágica, que a simple vista puede parecer colosal o caótica, resulta ser una rotonda altamente eficiente. Capaz de soportar un flujo de tráfico de hasta 6200 coches por hora.

Como en cualquier rotonda la prioridad de paso será para los vehículos que estén circulando dentro de la rotonda. La construcción de esta rotonda permite que un coche cruce la rotonda, pasando de una rotonda satélite a la rotonda principal, para ir luego a la rotonda satélite que más le convenga para salir de la intersección. Esto permite cruzar el conjunto de rotondas sin necesidad de dar un rodeo alrededor de la intersección. Además, para llegar a una salida, existirán diferentes rutas. Esto hace que los vehículos tengan opción de tomar la ruta menos congestionada, y de esta forma reducir la necesidad de “pelear” por el espacio.

Como se ve en la Figura 11, las rotondas satélites actuarían como rotondas convencionales, con prioridad a la derecha o prioridad en el anillo (al circular por la izquierda en Reino Unido, su prioridad a la derecha se corresponde con nuestra prioridad a la izquierda), donde deberán ser los vehículos que se aproximen a ellas las que cedan el paso. La rotonda principal, en la que recordemos que se circula en sentido antihorario, lo que favorece el movimiento de los vehículos entre las rotondas satélite. Esta rotonda funciona como una rotonda con prioridad a las entradas, donde los vehículos que circulan por dentro de la calzada anular cederán el paso a los vehículos que se aproximen a la calzada anular (de la rotonda principal). Este cambio en la prioridad, el paso de prioridad en el anillo en las rotondas satélite a prioridad en el anillo en la rotonda principal permite un movimiento fluido desde la salida de la rotonda satélite a través de la rotonda principal hasta otra rotonda satélite. La rotonda exterior, que en realidad podría considerarse que son carriles que conectan las rotondas satélite, aporta una vía alternativa alrededor de toda la maraña de rotondas. Esto permitirá que, en caso de accidente, o autobloqueo espontáneo debido a cualquier motivo, los vehículos dispongan de una vía alternativa para seguir circulando y evitar así atascos.

Por complicado que pueda parecer, en realidad, atravesar esta intersección sería tan simple como, mirar la salida a la que te quieras dirigir y avanzar hacia ella adaptándote a las curvas de las distintas rotondas. En caso de que haya un gran número de vehículos en por el trayecto que atraviesa la rotonda, existe la alternativa de bordear a través de la rotonda exterior.



*Figura 12. Ejemplo de rotonda mal planificada.*

Las intersecciones poco a poco van siendo sustituidas por rotondas, como ya hemos visto esto es debido a la mejora que supone en cuanto a flujo de tráfico y a la reducción de la peligrosidad. Para facilitar el diseño de una rotonda conviene utilizar simuladores para probar distintas variables antes de llevar a cabo la ejecución de la rotonda.

Un simulador es una herramienta extremadamente útil a la hora de diseñar una rotonda. Su ubicación, características, el tráfico de la zona, como funcionaria en determinadas condiciones, por ejemplo, si se cortara una calle, o un carril... Son todo variables que se han de tener en cuenta al planificar la construcción de una rotonda si no se quiere acabar estorbando al tráfico en lugar de ayudarlo. (Figura 12 [14])

Como la rotonda de la Figura 12, situada en un polígono industrial de Valverde de Majano, Segovia, una rotonda con una entrada y una salida, cuya única posible utilidad sería la de realizar un cambio de sentido. Función que ya estaba

suelta por la rotonda que se ve arriba en la foto, que además distribuye a los vehículos que llegan a la intersección.



*Figura 13. Rotonda innecesaria 1*



*Figura 14. Rotonda innecesaria 2*

En las Figuras 13 y 14 vemos otros ejemplos de rotondas que, a simple vista, parecen no cumplir ninguna función más que la de ser un gasto de dinero, de recursos y de tener un impacto ecológico negativo en la zona en la que se construyen. ¿Quizá su construcción obedece a un futuro proyecto? Quién sabe, solo aquellos que decidieron llevar a cabo semejantes obras.

Ya hemos visto lo necesario que es el diseño de una rotonda, pero ¿Cómo nos puede ayudar un simulador en esto? Vamos a verlo.



## 2.3 Herramientas para simulación de tráfico en rotondas

### 2.3.1 Traffic Flow de Martin Treiber

Como ya sabemos, las primeras rotondas que se proyectaron funcionaban según la norma de dar prioridad a las entradas, a diferencia de las rotondas tal y como las conocemos hoy en día, que dan prioridad al anillo. Vamos a comprobar si esto supone una mejora importante, para ello vamos a utilizar el siguiente simulador de Martin Treiber [15]

Este consta de distintos parámetros como el flujo total de vehículos por hora, carga de la vía principal, parámetros relativos a los vehículos, semáforos... (Figura 15)



Figura 15. Simulador

Como se puede ver en la Figura 15, podemos elegir si dar prioridad al anillo, a las entradas, o hacer que sea el primer vehículo en llegar a la intersección el que tenga prioridad de paso. Para esta prueba escogeremos la primera opción, prioridad al anillo. Después de 70s de tiempo de simulación, vemos que el flujo de vehículos es constante, sin problema alguno, tal y como debería funcionar una rotonda, Figura 16.





Figura 16. Prioridad al anillo 70 segundos

En cambio, si cambiamos la prioridad, para dársela a las entradas. Tras 70s de simulación, la rotonda se encuentra totalmente bloqueada.



Figura 17. Rotonda bloqueada

Con una simulación relativamente sencilla, hemos podido comprobar cómo el cambio de prioridad, de las entradas al anillo, mejora enormemente el funcionamiento de una rotonda.

Sin un simulador, el llevar a cabo una prueba de campo. El cambiar la prioridad de las entradas al anillo, supondría un problema enorme. Supondría cambiar las normas de circulación, por lo que sería necesaria una prueba en un circuito cerrado, con usuarios preparados para circular en estas nuevas circunstancias. En definitiva, sería una prueba costosa y difícil de ejecutar. Frente a la “facilidad” del simulador, la idea de realizar una prueba practica parece algo ilógico. Bien es cierto que los simuladores pueden presentar distintos niveles de precisión, pueden disponer de tantas variables como uno sea capaz de programar, la dificultad de prever cómo se comportaran los usuarios al llegar a la rotonda hace que sea extremadamente difícil crear un simulador que sea realmente fiel a la realidad. Sin embargo, un simulador tan “sencillo” como el de Martin Treiber, nos permite hacernos a la idea de la mejora que ha supuesto el cambio de prioridad en las rotondas.

### 2.3.2 Roundabout Simulator:

En el próximo ejemplo tenemos un simulador más simple aún que el anterior. Se llama *Roundabout Simulator* [16]. En este caso únicamente tendremos una rotonda inteligente en este caso como vemos en la Figura 18. Junto con el plano de la rotonda, tenemos un menú que nos permite seleccionar hacia donde circula el tráfico. Si seleccionamos que sea en sentido Norte, el simulador nos mostrará las posibles opciones que tendrá un vehículo que se aproxima a la rotonda desde el Sur para circular por el anillo.

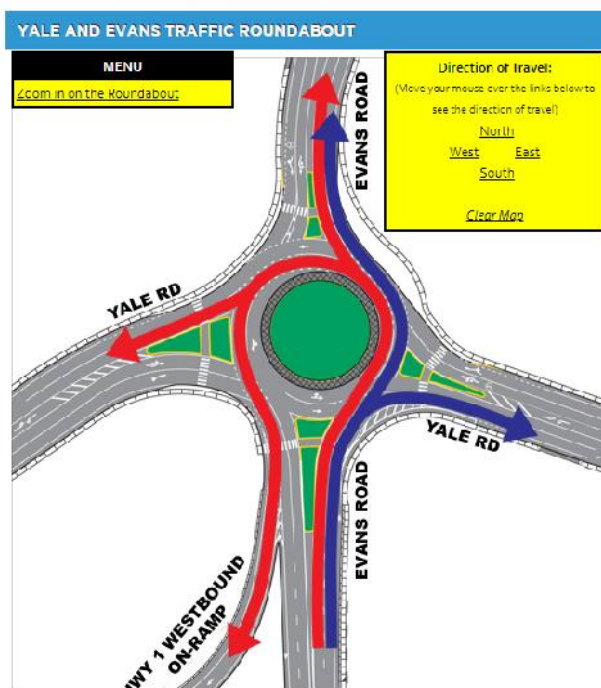
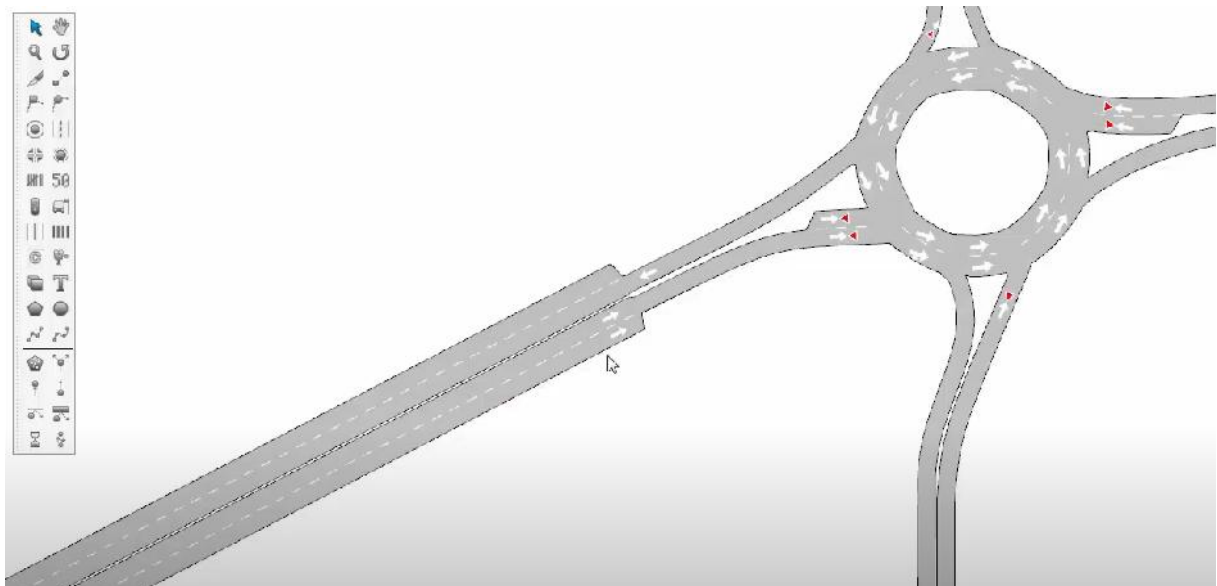


Figura 18. Roundabout Simulator

Este es un simulador extremadamente sencillo, cuya funcionalidad probablemente no vaya más allá de la de enseñar como deberían circular los vehículos en una rotonda.

### 2.3.3 Simulador de Aimsun:

Contrasta con este último simulador, el simulador adquirido por el Ayuntamiento de Burgos. Un programa creado por Aimsun y vendido al Ayuntamiento por 25.750 euros. Un simulador mucho más avanzado, comprado para realizar previsiones sobre la situación del tráfico en determinadas condiciones lo que ayudará en la toma de decisiones para mejorar la circulación en la ciudad.



*Figura 19. Sistema de control de tráfico mediante simulación. Aimsun.*

En la Figura 19 podemos ver cómo el simulador permite construir la rotonda y las vías circundantes. Se trata de un simulador altamente avanzado, donde podemos tener en cuenta todo tipo de variables.

### 2.3.4 Sumo:

El simulador SUMO (*Simulation of Urban MObility*) fue desarrollado por el centro aeroespacial de Alemania junto con la comunidad de usuarios. Está disponible de forma gratuita desde 2001. Como otros simuladores permite el diseño e implementación de estrategias de control del tráfico antes de utilizarlas en el mundo real. SUMO permite a sus usuarios realizar cambios en el código fuente del programa mediante una licencia de código abierto para experimentar con nuevas soluciones.





Figura 20. Plano general de una simulación en SUMO

SUMO pretende ser una herramienta capaz de simular cualquier elemento móvil existente en un tráfico: coches, autobuses incluyendo pasajeros, trenes, bicicletas, peatones e incluso barcos.

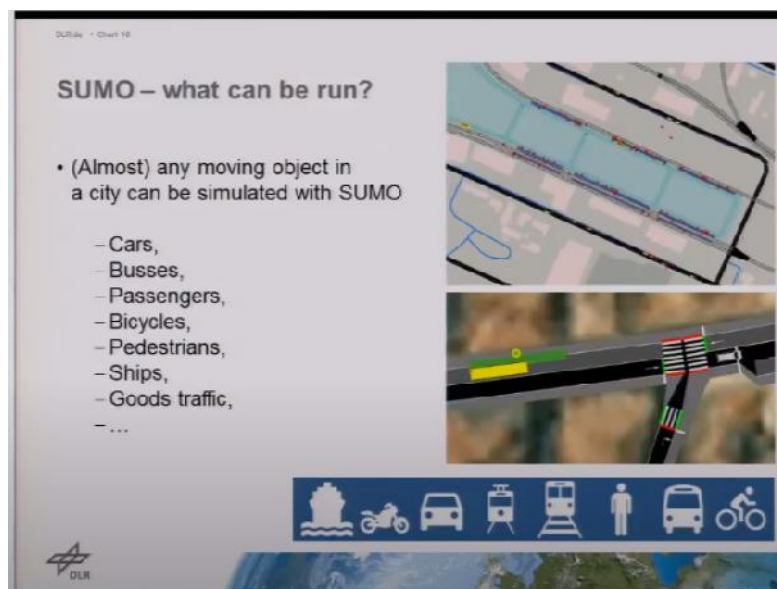
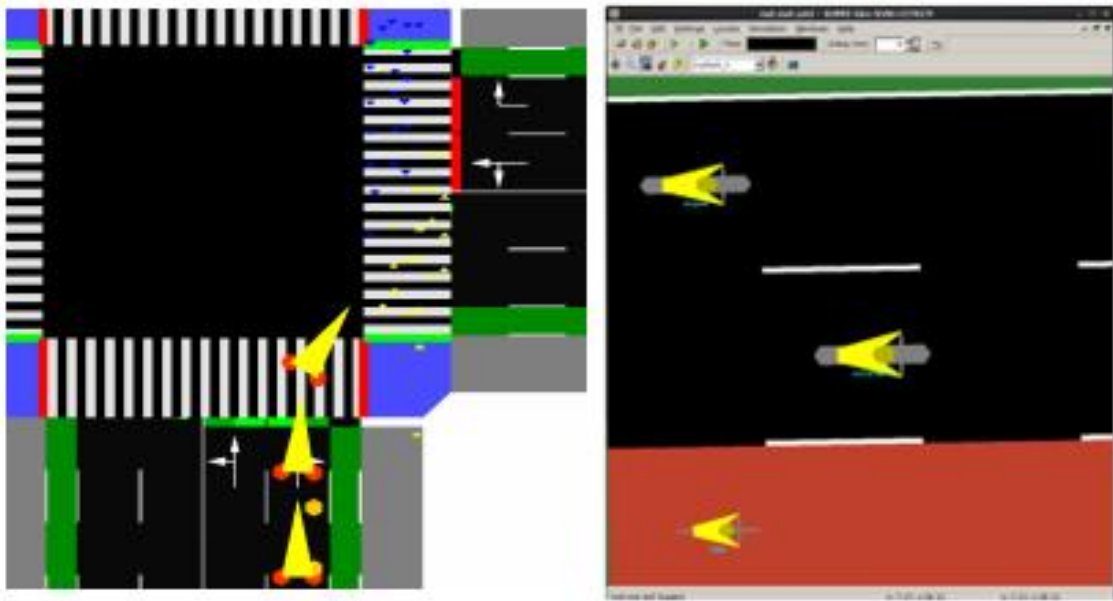


Figura 21. SUMO, ¿Qué puede simular?[17]

Para la simulación SUMO necesita 3 cosas:

- Infraestructura. La red de carreteras por la que se moverán los vehículos. Puede tratarse de las vías del tren, o de canales para el movimiento de barcos.
- Elementos de señalización. Semáforos, señales de circulación y las reglas que regulan el comportamiento del tráfico.
- Una demanda de transporte que de sentido a la circulación del tráfico.



*Figura 22. Intersección en SUMO (izquierda) Motocicletas (derecha)*

Como hemos visto se trata de un simulador altamente avanzado, de código abierto y con multitud de opciones de personalización. Permitiendo incluso una modificación de las normas de circulación desde 0, como podría ser cambiar el sentido de la circulación de la rotonda, o habilitar un carril para la circulación en sentido contrario al habitual dentro de la rotonda. Se trata de una opción interesante para la simulación, que podría considerarse para futuros proyectos.

#### 2.3.5 Matlab:

Para preparar nuestro simulador grafico de rotondas vamos a utilizar Matlab. Matlab se presenta como una plataforma diseñada específicamente para ingenieros y científicos. Éste ofrece una amplio abanico de opciones para la manipulación de matrices, lo que nos va a ayudar a manejar los datos durante la simulación.

Podemos crear diferentes algoritmos que nos ayuden a llevar a cabo nuestra simulación. Matlab tiene una funcionalidad integrada que nos ayudará a crear nuestra interfaz de usuario, llamada GUIDE (Graphic User Interface Development Environment).

Permite añadir cuadros de dialogo, distintos controles de interfaz de usuario, como botones y controles deslizantes. Esto nos permitirá representar gráficamente nuestra rotonda dentro de una interfaz que a la vez nos permitirá configurar la simulación y visualizar los resultados.

### 3. Análisis y diseño

Para llevar a cabo nuestro simulador en Matlab, hemos tomado como base el proyecto de Dominik Eugster & Roman Fuchs, de Zurich:

“Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB”

[http://webarchiv.ethz.ch/soms/teaching/MatlabSpring11/projects/Traffic\\_Flow-Two\\_lane\\_Roundabout\\_Eugster\\_Fuchs.pdf](http://webarchiv.ethz.ch/soms/teaching/MatlabSpring11/projects/Traffic_Flow-Two_lane_Roundabout_Eugster_Fuchs.pdf)

Se trata de un proyecto muy interesante, que resuelve problemas como el de la geometría de la rotonda, los coches dentro de esta, o las reglas de movimiento de los coches. Partir de este proyecto como base nos ha permitido acelerar el proceso de creación de la interfaz de nuestro simulador, y así poder centrarnos en las herramientas de análisis que utilizaremos para evaluar la viabilidad de distintas rotondas.

Hemos utilizado su geometría, una rotonda de dos carriles. En ella convergen 4 vías, de 3 carriles cada una. Tanto los carriles de las vías que conectan con la rotonda, como los carriles del anillo tienen 20 celdas.

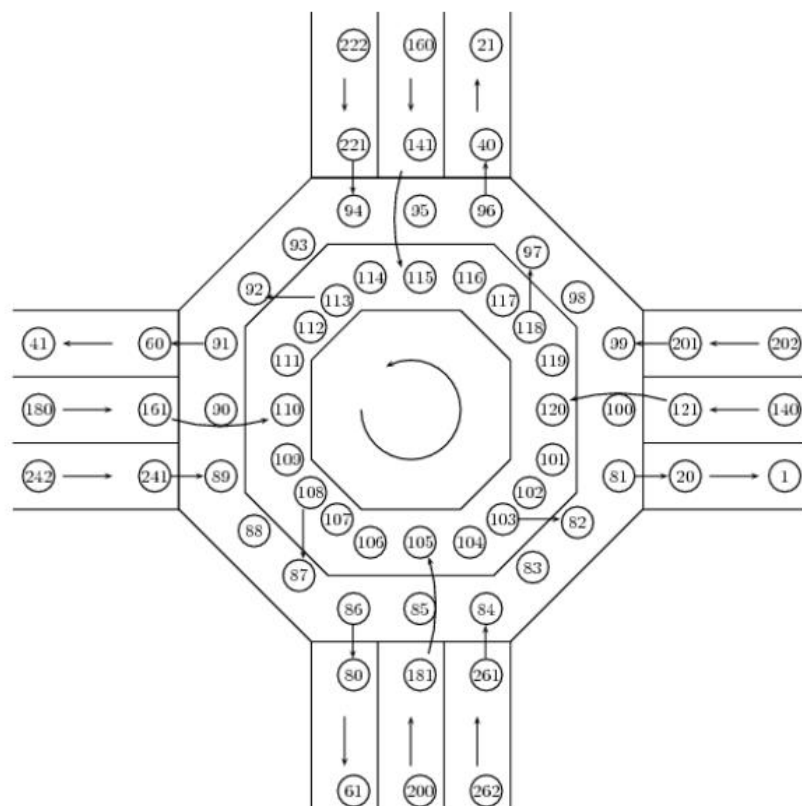


Figura 23. Boceto de la rotonda

Además de la geometría hemos utilizado la dinámica de circulación diseñada en su proyecto. Los vehículos avanzarán por las celdas de las vías, de una en una

en caso de que no haya otro vehículo en la celda siguiente, hasta llegar a donde estaría la línea de ceda el paso. Una vez aquí cederán el paso a los vehículos que circulen por el anillo interno. Cuando se disponga de un espacio suficiente (dos celdas libres) se incorporarán en el carril interno o en el exterior dependiendo de cuál sea su destino. Un vehículo que vaya a coger la primera salida utilizará el carril externo del anillo, mientras que si se dirige a alguna de las demás salidas entrará en el carril interno.

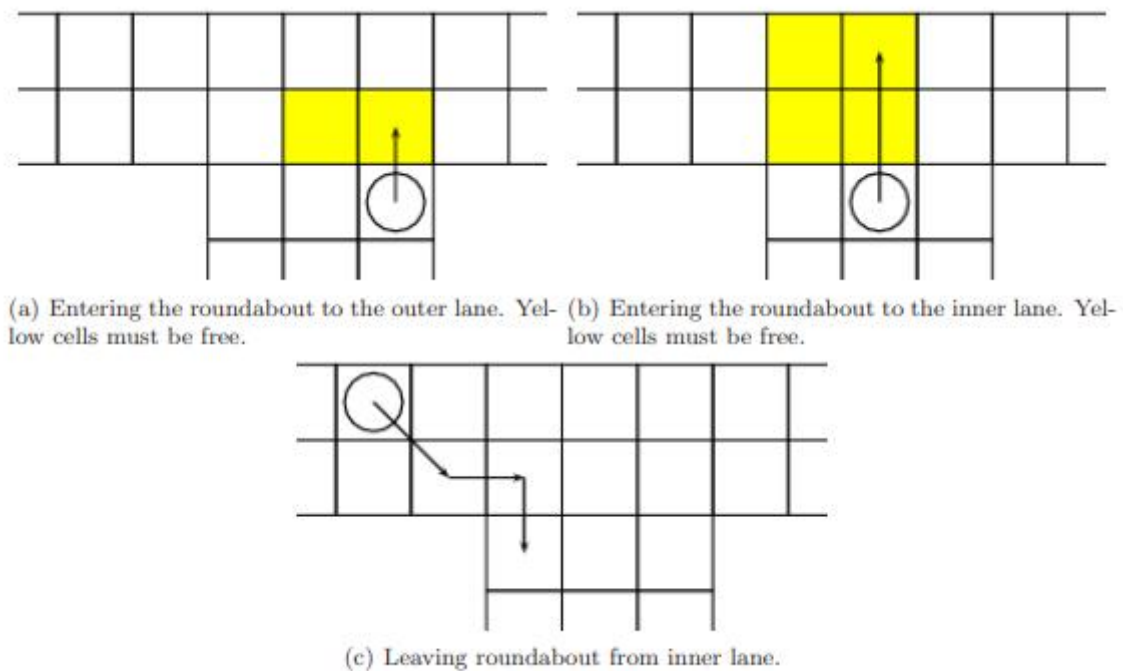


Figura 24. Incorporaciones y salidas

Los vehículos que circulen por el carril interno se cambiarán del carril interno al externo antes de llegar a su salida.

Con la rotonda, los coches, y las mecánicas de movimiento preparadas. Podremos empezar a estudiar distintos tipos de rotondas para ver su efecto sobre la fluidez del tráfico. Partiremos del caso más simple: una rotonda de dos carriles donde convergen 4 calles.

Después comprobaremos qué efectos tendrían distintas modificaciones como, por ejemplo:

- Añadir carriles laterales a la rotonda.
- La utilización de semáforos “inteligentes”.
- Usar un semáforo giratorio para dirigir el tráfico.
- Probar distintas variables del semáforo giratorio.

## 4. Implementación y desarrollo

En este capítulo se desarrollarán tres secciones, el diseño de una interfaz polar sobre la que ejecutar nuestro simulador, el análisis de la interfaz desarrollada por Fuchus de la que hablamos anteriormente, la implementación de herramientas sobre la interfaz Fuchus para buscar la optimización de las rotondas.

### 4.1 Versión Polar:

Primero vamos a detallar el diseño de una interfaz polar sobre la que ejecutar nuestro simulador. Dado que partimos de la hipótesis de que la rotonda estará en Europa, y concretamente en España, y que se circula principalmente por la derecha, en la rotonda los vehículos circularán en sentido antihorario. Los elementos que se incluirían en esta interfaz serían:

- **Vías de acceso:** Cuatro vías que convergen en la rotonda. Cada una de ellas con dos carriles, uno para cada sentido de la circulación.
- **Calzada anular:** Constituida por un carril para la circulación. Todos los vehículos circularán por ésta rodeando lo que sería la isleta central de la rotonda. El anillo tendrá distintas celdas, representadas como rombos verdes, tal y como se ve en la Figura 25.
- **Los vehículos:** Que llegarían por las vías hasta la calzada anular para incorporarse cuando dispongan de espacio suficiente. Todos los vehículos tendrán el mismo tamaño y velocidad.

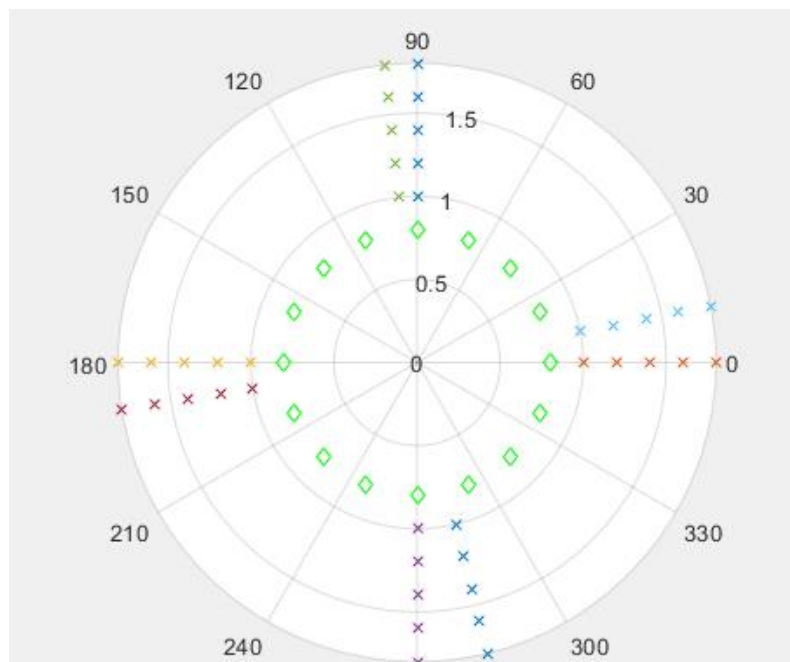


Figura 25. Concepto de la rotonda

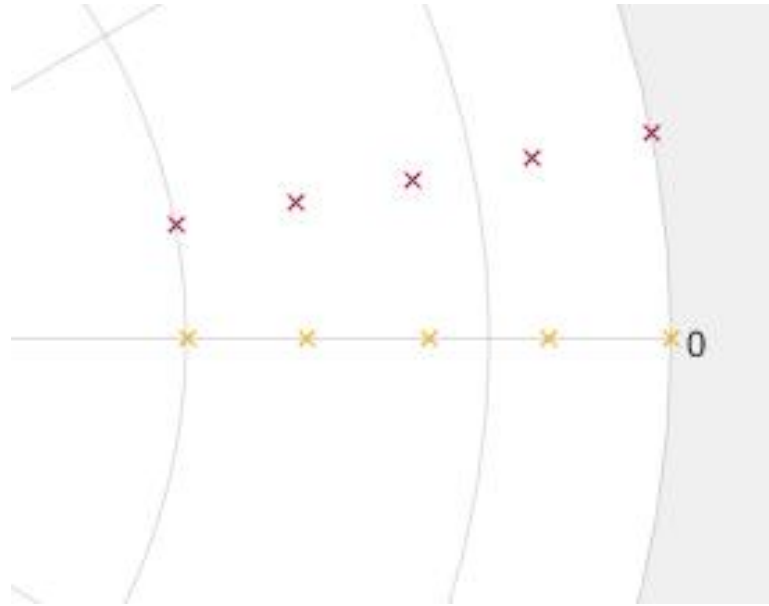


Figura 26. Detalle salida este

La idea es representar la rotonda mediante un gráfico polar. Cada una de las cuatro vías se representa en  $n \cdot 90$  grados, siendo  $n$  el número de la vía, que va desde 0 hasta 3. Todas las calzadas funcionan como matrices, en ellas cada celda se corresponderá con una celda de la matriz.

Para explicar el funcionamiento vamos a ver cómo funciona el proceso de salida de los vehículos. En la matriz  $G$  se almacena la información de todos los vehículos que hay en la rotonda, esta se corresponde con el carril exterior del anillo, la matriz  $G_i$  será la correspondiente al anillo interno. Si la rotonda tiene posiciones=40, la matriz  $G$  será de forma  $2 \times 40$ . En la Figura 27 vemos un ejemplo de 10 elementos de la matriz  $G$ . Las columnas representan las posiciones de la rotonda, la fila 1 sirve para guardar la información del destino de los vehículos y la fila 2 para guardar el origen de los vehículos de la forma siguiente: 1-Norte. 2-Oeste. 3-Sur. 4-Este. Un cero significa que la casilla está vacía y por tanto no hay ningún vehículo en ella. De esta forma, por ejemplo, en la posición 1 encontraríamos un vehículo que se dirige hacia el Sur (3) y que viene desde el Norte (1). En la 2, un vehículo que va hacia el Oeste desde el Norte.

	1	2	3	4	5	6	7	8	9	10
1	3	2	0	0	3	3	0	0	2	0
2	1	1	0	0	2	2	0	0	3	0

Figura 27. Matriz  $G$

Para esta prueba aparecerán vehículos desde las vías de entrada a la rotonda siempre que tengan la celda de entrada al anillo libre. Veamos el código correspondiente a la entrada de los vehículos.



```

% Todos los que entran Norte 270, 11 saltos
if G(1,1) == 0 % entrada
    G(:,1) = [salidaN(ceil(3*rand));1];
    Ne = Ne + 1; % contador entrada de vehículos Norte
end
% Todos los que entran W giran 180 hacia W , 7 saltos
if (G(1,5) == 0) % entrada vehículo por W
    G(:,5) = [salidaW(ceil(3*rand));2];
    We = We + 1; % contador entrada vehículos W
end
% Todos los que entran Sur giran 90 , 3 saltos
if G(1,9) == 0 % entrada de vehículo por S
    G(:,9) = [salidaS(ceil(3*rand));3];
    Se = Se + 1; % contador entrada vehículos Sur
end
% Todos los que entran Sur giran 180 , 7 saltos
if G(1,13) == 0 % entrada de vehículo por E
    G(:,13) = [salidaE(ceil(3*rand));4];
    Ee = Ee + 1; % contador entrada vehículos Este
end

```

Si alguna de las casillas de entrada (1, 5, 9, 13) están vacías, es decir a 0. Se rellena las filas de la matriz en la columna correspondiente. Para un vehículo que entre desde el Norte el numero de la fila 2, la fila del origen del vehículo se rellena con un 1, será un 2 para el Oeste, 3 para el Sur y 4 para el Este. El número del destino, el de la fila 1, se obtiene eligiendo de forma aleatoria un numero de los arrays que contienen los posibles destinos. Estos se definen en el archivo Glorieta.m:

```

salidaN=[2,3,4];
salidaW=[1,3,4];
salidaS=[1,2,4];
salidaE=[1,2,3];

```

De forma que los vehículos que entren desde el Norte podrán dirigirse a cualquier salida menos a la Norte, es decir que el cambio de sentido no se contempla en este caso. Por esto el array salidaN no tiene la opción 1. El mismo razonamiento se aplicará a los demás arrays de las demás salidas.

Para simular el movimiento de los vehículos dentro de la rotonda, se tienen dos matrices auxiliares, Gt y Gti. En ellas se almacena el estado actual de las matrices G y Gi desplazado, para luego recuperarlo en las matrices principales, veamos el código.

```

for giro=1:1:posiciones
    if giro==1 Gt(:,giro)=G(:,posiciones);
                Gti(:,giro)=Gi(:,posiciones);
    else
        Gt(:,giro)= G(:,giro-1);
        Gti(:,giro)= Gi(:,giro-1);
    end
end

```

Como queremos desplazar la posición de todas las celdas de las matrices G y Gi una posición hacia la derecha, guardaremos en Gt(:,2) el estado actual de G(:,1) y así hasta recorrer toda la matriz. Como intentar guardar de esta forma

en  $Gt(:,1)$  la información de  $Gt(:,0)$  resultaría en un error, vamos a crear una excepción para el caso en el que  $giro=1$  ( $giro$  es la variable que utilizaremos para recorrer las matrices). Para el caso  $giro=1$ , cogeremos la información del final de la matriz, es decir utilizaremos la variable  $posiciones$ , que tiene el número de columnas totales de la matriz  $G$ .

Para las salidas el proceso será similar a las entradas:

```
if (G(1,posiciones) == 1 || Gi(1,posiciones) == 1 )
    G(:,posiciones) = -1;
    Ns = Ns + 1; % salida vehículo a Norte, incr- contador salida
    vehículos Norte
else
end
if (G(1,4) == 2 || Gi(1,4) == 2) % salida de vehículo a W
    G(:,4) = 0;
    Ws = Ws + 1 ;% contador salida vehículos W
end
if (G(1,8) == 3 || Gi(1,8) == 3) % salida de vehículo a S
    G(:,8) = 0;
    Ss = Ss + 1 ;% contador salida vehículos Sur
end
if (G(1,12) == 4 || Gi(1,12) == 4) % salida de vehículo a E
    G(:,12) = 0;
    Es = Es + 1 ;% contador salida vehículos Este
end
```

Las posiciones para las salidas son las celdas (16, 4, 8, 12). Recordamos que en la variable  $posición$  se almacena el número total de celdas que tiene la calzada anular. De forma que la salida correspondiente a la vía Norte será la última celda.

Para representar gráficamente el origen y el destino de los vehículos en la rotonda, vamos a identificar cada vehículo con una forma geométrica (Norte-estrella, Oeste-cuadrado, Sur-rombo, Este-círculo) y con un color. El color coincidirá con la vía de origen y la forma geométrica coincidirá con la salida hacia la que se dirige. Para facilitar el entendimiento de la representación, se representan las vías de acceso y salida de la rotonda con la misma simbología que los vehículos. Lo vemos en la Figura 28. De un vistazo podemos identificar el origen y el destino hacia el que se dirigen los vehículos. Una estrella magenta



representa un vehículo que ha entrado desde el Este y que se dirige hacia el Norte, un círculo verde habrá entrado por el Oeste y se dirige hacia el Este, etc.

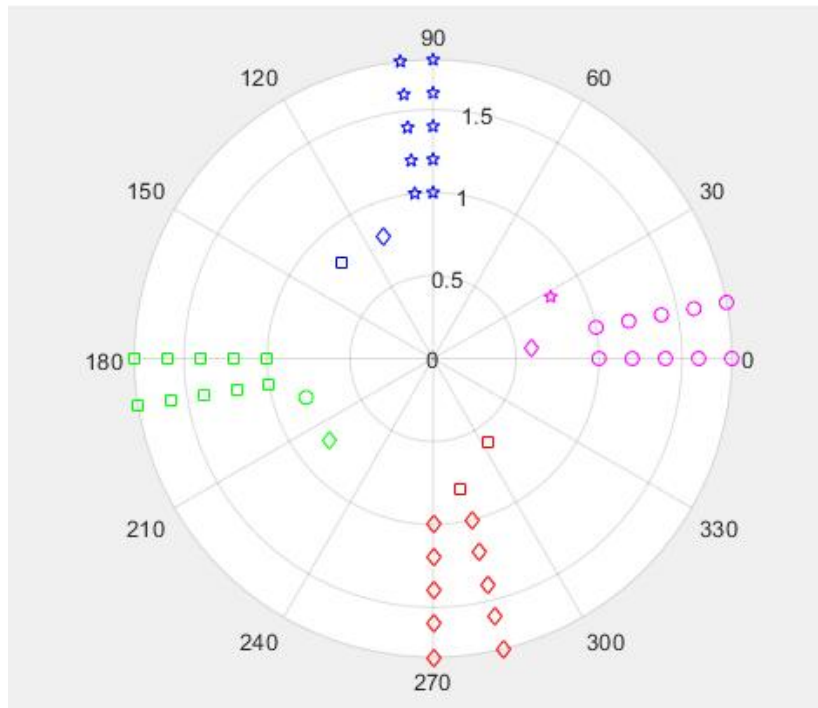


Figura 28. Rotonda con vehículos y representación de colores y formas.

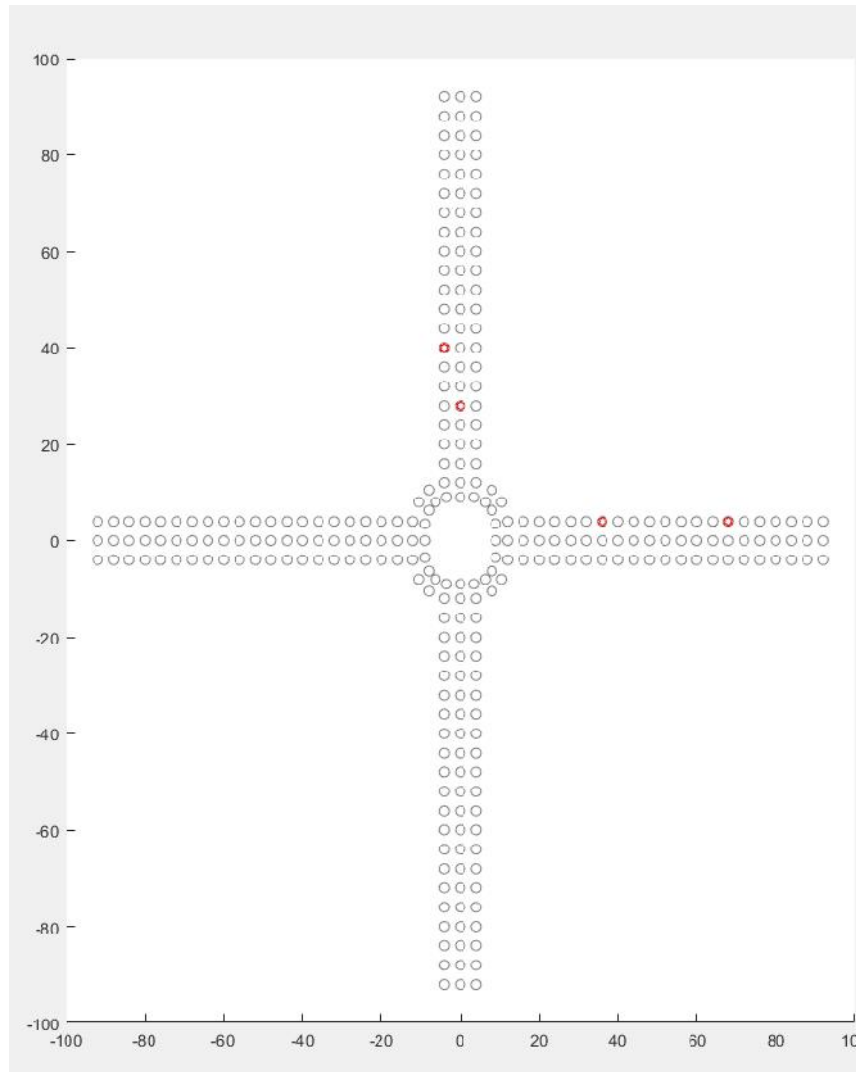
Con esto hemos conseguido una forma de representar los vehículos dentro de la rotonda, su origen y su destino. La representación polar ofrece cierta comodidad a la hora de representar las calzadas anulares, sin embargo, la representación de las demás calzadas en un diagrama polar resulta no ser la mejor opción. A partir de ahora, cambiaremos nuestro eje de coordenadas a uno cartesiano.

#### 4.2 Análisis de la plataforma de Fuchus

Tal y como se expuso previamente, el siguiente paso fue tomar como proyecto base de referencia el “Lecture with Computer Exercises: Modelling and Simulating Social Systems with MATLAB” de Dominik Eugster & Roman Fuchs, que ya fue presentado en Capítulos anteriores.

El funcionamiento será similar al que teníamos antes. Los vehículos aparecerán al principio de las vías y desaparecerán al llegar al final de estas. Cada vía consta de 2 carriles de entrada a la rotonda y de uno de salida. Gráficamente tenemos las calzadas laterales representadas por circunferencias en un eje cartesiano, cada circunferencia será una posición en la que podrá haber un vehículo. La

calzada anular también está representada con circunferencias. Los vehículos aparecerán como círculos rojos dentro de las circunferencias.



*Figura 29. Representación cartesiana*

Vamos a ver más de forma más detallada como está creada la rotonda. Para ello, vamos a volver a la Figura 23, donde tenemos el esquema de la rotonda, con los números que identifican cada celda.

Como se puede ver en la Figura 23, cada calzada se corresponde con un array de 20 celdas. Por ejemplo, en la calzada Este, en el carril de salida de la rotonda tendremos las celdas 1-20. Siendo la 20 la más próxima a la rotonda, y la 1 la más alejada de esta. Las celdas 21-40 son las del carril de salida de la calzada Norte en orden descendente desde el anillo, igual que la salida Este. Las siguientes celdas, 41-60 y 61-80, están en los carriles de salida Oeste y Sur respectivamente. Las celdas 81-100 y 101-120 constituyen los carriles anulares de la rotonda, siendo el primero el carril externo y el segundo el carril interno. Finalmente, los carriles de aproximación izquierdos a la rotonda tendrán las celdas desde la 121 hasta la 200, siendo el orden de las celdas descendente desde la calzada anular, empiezan en la vía Este y aumentan en sentido

antihorario. Esto es, 121-140 Este, 141-160 Norte, 161-180 Oeste, 181-200 Sur. En los carriles de aproximación derechos están las celdas 201-280. En la Figura 23 se puede ver que hay un error, cada carril son 20 celdas, sin embargo, en la Figura 23 está indicado que hay solo 1 celda: 201-202, todos los carriles de aproximación derechos tienen los mismos errores. Esta información se inicializa en la función **initGeom.m** y se guarda en la matriz **cases**. La matriz **cases** es una matriz 280x6:

- La primera columna **cases(:,1)** tiene el numero identifi cativo del vehículo que la ocupa si hay alguno, si está vacía habrá un 0.
- La segunda columna **cases(:,2)** contiene la coordenada X del centro de la circunferencia que representa la posición.
- La tercera columna **cases(:,3)** son las coordenadas Y del centro de la circunferencia de esa posición de la rotonda.
- Las columnas 3-6 se mantienen a 0 y no se utilizan.

	1	2	3	4	5	6
1	0	92	-4	0	0	0
2	0	88	-4	0	0	0
3	0	84	-4	0	0	0
4	0	80	-4	0	0	0
5	0	76	-4	0	0	0
6	0	72	-4	0	0	0
7	0	68	-4	0	0	0
8	0	64	-4	0	0	0
9	0	60	-4	0	0	0
10	0	56	-4	0	0	0
11	0	52	-4	0	0	0
12	0	48	-4	0	0	0
13	0	44	-4	0	0	0
14	0	40	-4	0	0	0
15	0	36	-4	0	0	0

Figura 30. Muestra matriz **cases(1:15, :)**

Inicialización de la rotonda, **initGeom.m**. Veamos la inicialización de las posiciones de la rotonda.

```
cases=zeros(280,6)

lane1=92:-4:16;
lane2=16:4:92;
v = [0 4 8 10.5 12]';
cases(81:85,2:3)=[sort(v,'descend'), [-4 -8 -10.5 -12 -12]]';
cases(86,2:3)=[-4, -12];
cases(87,2:3)=[-8, -10.5];
cases(88,2:3)=[-10.5, -8];
cases(89,2:3)=[-12,-4];
cases(90,2:3)=[-12, 0];
```

```

cases(91:100,2:3)=-cases(81:90,2:3);

cases(101,2:3)=[9,-3.5];
cases(102,2:3)=[8,-6.3];
cases(103,2:3)=[6.3,-8];
cases(104,2:3)=[3.5,-9];
cases(105,2:3)=[0,-9];
cases(106,2:3)=[-3.5,-9];
cases(107,2:3)=[-6.3,-8];
cases(108,2:3)=[-8,-6.3];
cases(109,2:3)=[-9,-3.5];
cases(110,2:3)=[-9,0];
cases(111:120,2:3)=-cases(101:110,2:3);

```

Primero se reserva la memoria para la matriz **cases** creando una matriz de ceros. **lane1** y **lane2** son las coordenadas de las calzadas laterales de la rotonda, para obtener las coordenadas del lado opuesto se invierten, **-lane1** y **-lane2**. Luego están las coordenadas de las distintas celdas que forman los carriles de la calzada anular **cases(81:100)** calzada exterior, **cases(101:120)** la calzada interior. A continuación, tenemos las coordenadas de las vías de entrada y salida a la rotonda.

```

cases(121:140,2)=lane2;      %carril entrada ESTE
cases(141:160,3)=lane2;      %carril entrada NORTE
cases(161:180,2)=-lane2;     %carril entrada OESTE
cases(181:200,3)=-lane2;     %carril entrada SUR
%-----CALZADA ESTE
cases(201:220,2)=lane2; %coordenada X
cases(201:220,3)=4;      %coordenada Y
cases(1:20,2)=lane1;
cases(1:20,3)=-4;
%-----CALZADA NORTE
cases(221:240,2)=-4;      %coordenada X
cases(221:240,3)=lane2; %coordenada Y
cases(21:40,2)=4;
cases(21:40,3)=lane1;
%-----CALZADA OESTE
cases(241:260,2)=-lane2;%coordenada X
cases(241:260,3)=-4;      %coordenada Y
cases(41:60,2)=-lane1;
cases(41:60,3)=4;
%-----CALZADA SUR
cases(261:280,2)=4;      %coordenada X
cases(261:280,3)=-lane2;%coordenada Y
cases(61:80,2)=-4;
cases(61:80,3)=-lane1;

```

En este caso la matriz que guarda la información sobre los coches es **cars**. La matriz **cars** se inicializa una vez al iniciar la ejecución del programa en la función **initCars.m**. Como el número de coches por simulación suele ser bastante alto vamos a ver una pequeña muestra en la Figura 31. La estructura de la matriz **cars(x,9)** donde x será el número de coches es como sigue:

- Cada fila se corresponde con un vehículo.
- La primera columna **cars(:,1)** es la posición actual del coche. Este número identifica directamente el número de la celda en la que se sitúa.

- La columna **`cars(:,2)`** contiene el número de la posición anterior a la actual.
- Tercera columna, **`cars(:,3)`** identifica el origen del coche con un número de 1 a 4. Siendo 1 el Este y aumentando en sentido antihorario de forma que el 4 sea el Sur.
- Columna **`cars(:,4)`** es el número que identifica el destino del coche. Los números se corresponden con los de la columna de origen. (1-4; Este-Sur)
- La columna **`cars(:,5)`** funciona como un flag para saber si el coche está en espera. Si no pudiera avanzar a la siguiente celda que le toca porque hay otro vehículo en ella el flag se pondrá en 1(en cola). Mientras no se obstaculice su movimiento el flag se mantendrá en 0 (no en cola).
- La columna **`cars(:,6)`** almacena el tiempo en el que el coche entra a la rotonda. Cuando el tiempo de simulación coincide con el tiempo que aparece en este elemento de la matriz, el coche entra en la rotonda. También se utiliza para calcular el tiempo total que tarda el coche en atravesar la intersección.
- La columna **`cars(:,7)`** es un indicador del estado del vehículo. 1 significa que el coche aún no ha entrado en la rotonda, el tiempo de simulación será menor que el tiempo de entrada a la rotonda, columna 6. Un 0 indica que el coche está conduciendo, ha entrado en la rotonda, pero aún no ha llegado a su destino. Será un 2 cuando haya completado su trayecto y haya llegado a su destino.
- La columna **`cars(:,8)`** identifica por donde ha entrado el coche a la rotonda. Almacena la celda donde se inicia la trayectoria, por ejemplo, un coche que entre por el Este por el carril central inicia su trayectoria en la celda 140.
- La columna **`cars(:,9)`** guarda el valor de tiempo en que el coche llega a su destino, este dependerá de las condiciones del tráfico que tenga la rotonda. Se utiliza para calcular el tiempo que ha tardado el coche en llegar a su destino.

	1	2	3	4	5	6	7	8	9
1	173	174	3	2	0	1	0	180	0
2	253	254	3	1	0	1	0	260	0
3	157	158	2	1	0	5	0	160	0
4	277	278	4	2	0	5	0	280	0
5	259	260	3	1	0	7	0	260	0
6	260	0	3	4	0	7	0	260	0
7	159	160	2	1	0	7	0	160	0
8	220	0	1	3	0	8	0	220	0
9	240	0	2	3	0	8	0	240	0
10	0	0	1	3	0	8	1	220	0
11	0	0	3	1	0	8	1	260	0
12	0	0	3	1	0	8	1	260	0
13	0	0	1	3	0	9	1	220	0
14	0	0	1	3	0	11	1	220	0
15	0	0	2	1	0	12	1	160	0
16	0	0	4	3	0	13	1	200	0
17	0	0	1	3	0	13	1	220	0
18	0	0	3	2	0	14	1	180	0
19	0	0	1	3	0	15	1	220	0
20	0	0	3	1	0	15	1	260	0
21	0	0	2	1	0	15	1	160	0
22	0	0	3	1	0	16	1	260	0
23	0	0	3	1	0	16	1	260	0
24	0	0	1	3	0	18	1	220	0
25	0	0	1	2	0	19	1	220	0
26	0	0	3	1	0	19	1	260	0
27	0	0	3	4	0	19	1	260	0
28	0	0	1	4	0	20	1	140	0
29	0	0	4	1	0	20	1	280	0
30	0	0	4	2	0	21	1	280	0
31	0	0	1	3	0	22	1	220	0
32	0	0	4	1	0	22	1	280	0
33	0	0	3	1	0	22	1	260	0
34	0	0	1	3	0	22	1	220	0
35	0	0	1	2	0	23	1	220	0
36	0	0	4	2	0	23	1	280	0
37	0	0	1	3	0	23	1	220	0
38	0	0	1	3	0	24	1	220	0

Figura 31. Muestra matriz cars(1:38,:)

En la Figura 31 podemos ver como la matriz cars nos permite analizar el estado de los coches que hay o habrá en la rotonda. Por ejemplo, el coche 9 (fila 9) vemos que está ubicado actualmente en la celda 240, su posición anterior es la 0, por lo tanto, acaba de acceder a la rotonda. Ha entrado por el Norte (2) y se dirige al Este (3). En la columna 0 el flag está a 0, así que el coche no tiene ningún obstáculo delante y puede seguir circulando. El tiempo en el que el coche entró a la rotonda fue el 8. El 0 de la columna 7 indica que el coche está en mitad de su trayecto. En la columna 8 está la celda por la que entro a la rotonda. En la columna 9 se guardará el tiempo en el que el coche llegue a su destino, mientras tanto sigue a 0.

En la fila 10 se puede ver un coche que va a entrar en la siguiente iteración del bucle, ya que el tiempo de entrada es el mismo que el de la fila 9.

La matriz **cars** se inicializa una única vez, de forma aleatoria en parte. La matriz **freg** permite definir el porcentaje de tráfico que llegara a la rotonda de la siguiente forma:

```
freq = ...
[35 15 70 15;...
 15 40 20 40;...
 35 15 70 15;...
 15 40 20 40];
```

- La primera columna define el porcentaje de tráfico que llegara por desde cada vía. La primera fila se corresponde con el Este, la siguiente con el Norte etc.
- La segunda columna es el porcentaje de tráfico que tomara la primera salida.
- La tercera columna es el porcentaje de vehículos que seguirán recto, es decir tomaran la segunda salida.
- La cuarta columna es el porcentaje de vehículos que giraran a la izquierda, o tomaran la tercera salida.

Vamos a ver un fragmento de la función **initCars.m**, para explicar su funcionamiento:

```
cars=zeros(totalCars, 9);
randnbs=floor(tmax*rand(totalCars, 1));
randnbs=sort(randnbs);
cars(:,6)=randnbs;
cars(:,7)=ones(totalCars,1);
for i=1:totalCars
    if randnbs(i,1)<=freq(1,1) % Origin 1
        cars(i,3)=1;
        if randnbs(i,2)<= freq(1,2)
            cars(i,4)=2;
            cars(i,8)=220;
        elseif randnbs(i,2)<= freq(1,2)+freq(1,3)
            cars(i,4)=3;
```

Como se puede ver la inicialización dependerá de una matriz generada en función de varios parámetros, como **tmax** (tiempo máximo), **totalCars** (número total de coches) y de la matriz **freq**. Se rellena la columna 6 de **cars** con valores aleatorios entre 0 y **tmax** para utilizarlos como tiempo de entrada de los coches. Luego se decide de forma aleatoria, comparando valores en la matriz **randnbs** y **freq**, el origen y el destino de cada coche, así como si entrará por el carril derecho o el izquierdo. Para ver el código completo ir al Anexo 2.

La dinámica de movimiento de los coches está definida por 4 funciones:

**moveCarFromCross.m**, **moveCarOuterRoundabout.m**, **moveCarInnerRoundabout.m**, **moveCarToCross.m**.

En primer lugar, se ejecuta **moveCarFromCross.m**. Primero vacía los carriles de salida, esto lo hace comprobando si hay algún coche en las celdas 1, 21, 41, 61, si es así, se pone la columna 1 de **cars** a 0 (esta columna contenía la ID del coche que estaba en esa posición). La columna 7 de **cars** se cambia a 2, como ya hemos visto esto significa que el coche ha llegado a su destino. Y en la columna 9 de **cars** se guarda el tiempo en el que el coche ha salido de la rotonda. La función completa sería:

```

función [cars, cases] = moveCarFromCross.m(cars, cases, time)
for caseID = 1 : 80
    if (caseID==1 || caseID==21 || caseID==41 || caseID==61) %empty the
exiting cars
        if cases(caseID,1)>0
            cars(cases(caseID,1),7)=2;
            cars(cases(caseID,1),9)=time;
            cases(caseID,1)=0;
        end
    elseif (cases(caseID,1) > 0) % if there is a car in this case number
        thisCar=cases(caseID,1);
        if(cases(caseID-1,1)==0) % if next case empty, drive forward
            cars(thisCar,2)=caseID; %last position
            cars(thisCar,1)=caseID-1; % actual position
            cars(thisCar,5)=0; %not queuing
            cases(caseID-1,1)=thisCar;
            cases(caseID,1)=0;
        elseif cars(thisCar,2)==cars(thisCar,1)
            cars(thisCar,5)=1; %queuing
        else
            cars(thisCar,2)=caseID; %not yet queuing
        end
    end
end
end

```

Después de eliminar los coches que están en el extremo de las vías de salida, se busca un coche que haya en una de estas vías. Cuando encuentra uno comprueba si tiene algún coche en la celda siguiente (el orden en estas vías es descendente luego la celda siguiente la obtenemos restando 1 a la celda actual). Si no hay ningún coche se actualizan los valores correspondientes en cars y cases, serían:

- **Cars:** columnas 1,2,5. Respectivamente la posición actual, posición anterior y el flag no está en cola, por si se encontraba en cola.
- **Cases:** columna 1. Número de ID del coche actual.

Si se encuentra algún coche en la celda siguiente, simplemente se actualizará el flag de espera del coche, esto es, **cars(thisCar, 5)=1**, coche en espera. Cuando la celda siguiente se vacíe el coche podrá avanzar.

Después se ejecuta **moveCarOuterRoundabout.m**. En ella encontramos un switch **caseID**, que se repite con los valores de las celdas del carril exterior del anillo, es decir desde 81 hasta 100. Cuando se encuentra un coche en una de las salidas, y esa salida se corresponde con su destino: 81-Este, 84-Sur, 91-Oeste, 96-Norte, el coche “saldrá” de la rotonda. Por ejemplo, un coche cuyo valor en **cars(thisCar, 4)** sea 1, significará (como ya se ha explicado anteriormente) que su destino es la salida Este. Por lo tanto, circulará a lo largo del carril exterior, en cada iteración su posición se reduce en 1, ya que este carril tiene sus celdas numeradas desde 100 en la entrada Este, hasta 81 en la salida Este. Cuando llegue a la celda 81, pasará a la celda 20 (la primera del carril de salida Este), el proceso de salida consiste en actualizar las posiciones actual y anterior, bloquear la celda nueva (celda 20), liberar la celda anterior (81) y asegurarse de que el flag de espera del coche esta desactivado.



Como no todos los coches saldrán por el Este, es decir la celda 81, que es la más baja del carril exterior. Tiene que ser posible continuar circulando por la rotonda, es decir, pasar de la última celda del carril (81) a la primera (100). Esto se hace siguiendo el mismo proceso que con los coches que salen de la rotonda en esta salida, pero que si su destino (`cars(thisCar, 4)`) es distinto de 1, la celda nueva será la numero 100.

Hay que tener en cuenta que siempre antes de realizar un movimiento de un coche, se comprueba que la celda a la que se va a mover está vacía. De no estarlo se actualiza el flag en espera y se espera a que se libere la celda.

El código se puede consultar en el Anexo 2.

Tras mover los coches del carril exterior del anillo se moverán los de la calzada interna, con la función **`moveCarInnerRoundabout.m`**. El funcionamiento de esta será similar a la función encargada del carril exterior. Las celdas están numeradas desde 101 hasta 120, empezando igual que en el caso anterior desde el Este y acabando en la salida Este. Las salidas del carril en este caso no coinciden exactamente con las del anterior. Los coches que circulan por este carril deberán cambiarse al carril exterior con tiempo para poder dirigirse a la siguiente salida con seguridad, por lo tanto, existe un desplazamiento respecto a las salidas del carril externo. Salidas: 103-Este, 118-Norte, 113-Oeste, 108-Sur. El funcionamiento será igual que en el caso anterior. Los coches que salgan de este carril llegaran al carril exterior.

Por último, se ejecuta la función **`moveCarToCross.m`**. Esta función controla los coches que entran a la rotonda y los que llegan a los carriles laterales. Recorre las posiciones restantes de la rotonda, desde la 121 hasta la 280, con un switch case se comprueba si hay algún coche en las posiciones de entrada, que son:

- Este: 121 (izquierda) a 120 (interior). 201 (derecha) a 99 (exterior).
- Norte: 141 (izquierda) a 115 (interior). 221 (derecha) a 94 (exterior).
- Oeste: 161 (izquierda) a 110 (interior). 241 (derecha) a 89 (exterior).
- Sur: 181 (izquierda) a 105 (interior). 261 (derecha) a 84 (exterior).

Los vehículos que llegan por el carril izquierdo de una vía se entienden que no van a coger la primera salida de la rotonda, por lo que entran en el carril interno de la rotonda.

Durante la simulación, se representan los movimientos de los vehículos. Cada vez que un vehículo avanza una posición se borra y se vuelve a dibujar. De forma que de la sensación de que el círculo que representa el coche va avanzando por la rotonda. Además, se representa una gráfica con el número de coches que hay en espera.

Vamos a ver el resultado de la ejecución del programa completo. Para empezar, configuramos para que pase 1 único coche por la rotonda.

```
totalCars = 1;  
tmax = 5;
```

Tras la inicialización, la matriz cars queda como se ve en la Figura 32

	1	2	3	4	5	6	7	8	9
1	0	0	4	2	0	0	1	280	0

Figura 32. cars 1 coche

Como ya sabemos tras inicializar la matriz queda definido el momento en el que el coche aparecerá en la rotonda, así como por donde va a entrar a ella, y su destino. En este caso sabemos que va a entrar desde el Sur y se dirigirá al Norte, también sabemos que aparecerá en  $t=1$  de la simulación.

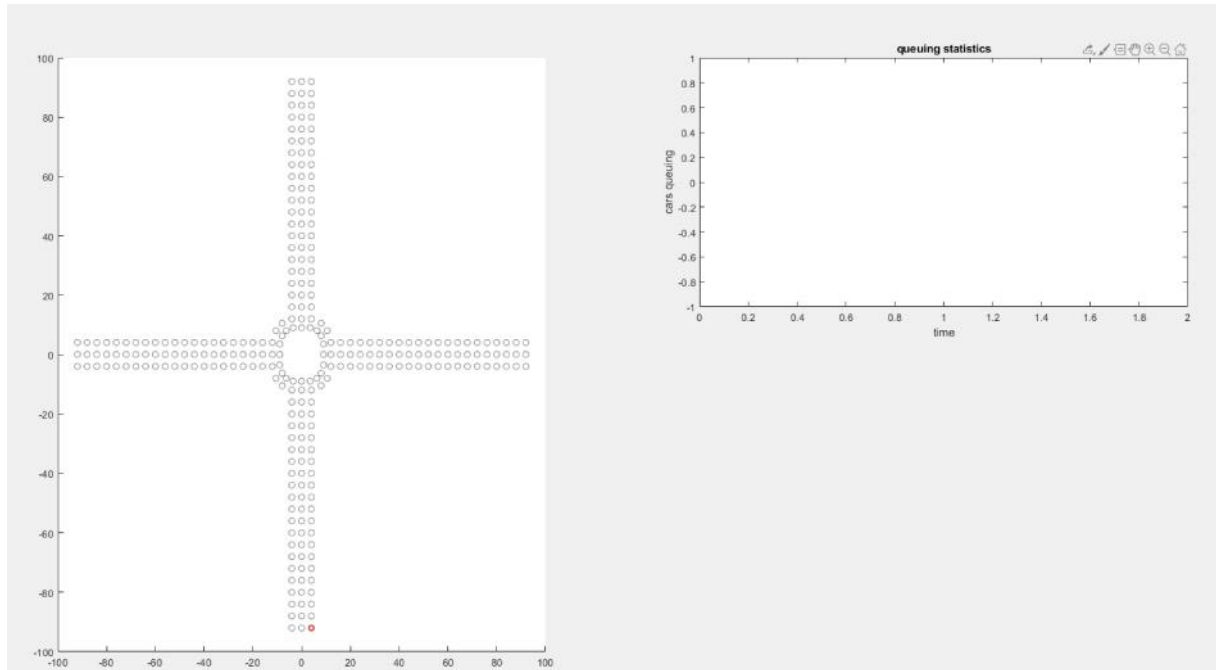


Figura 33. Output simulación 1 coche

En  $t=1$  el coche entra en la rotonda por el Sur, por el carril derecho. Se puede comprobar que en efecto el círculo rojo que representa el coche está en la celda 280. Durante la simulación el coche avanzará por el carril derecho hasta llegar al anillo, donde pasará al carril exterior para bordear la rotonda hasta llegar a su salida, en este caso en el Norte, a la celda 96 de donde pasará a la celda 40. Podemos ver el trayecto en la Figura 34.

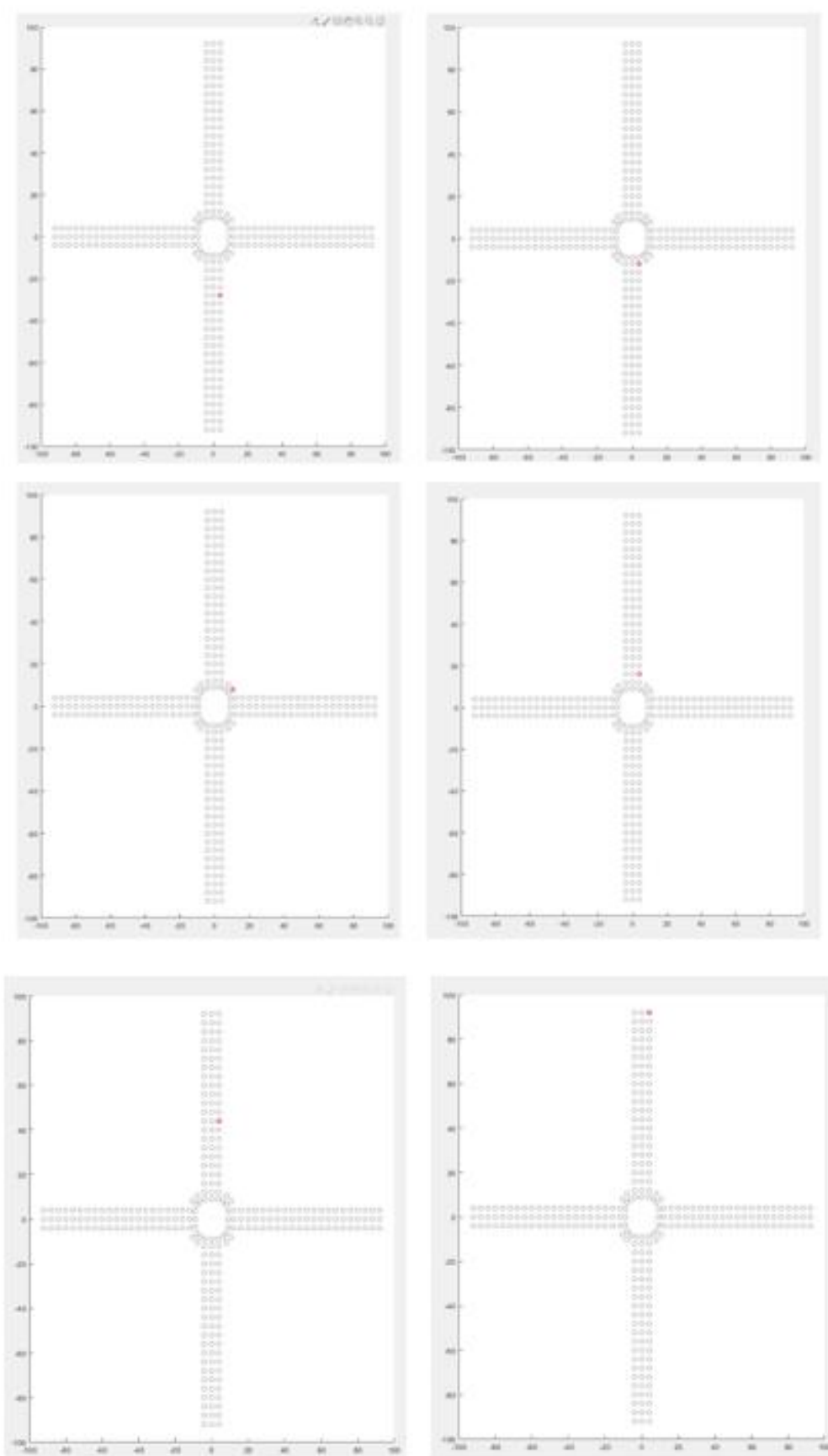


Figura 34. Trayectoria del coche a través de la rotonda

Dado que únicamente teníamos un coche circulando por la rotonda, la gráfica que representa los coches en espera no mostrará ninguna información útil, se mantendrá a 0 toda la ejecución.

Vamos a repetir este proceso, cambiando la configuración del tiempo y el número de coches, buscando provocar la saturación de la rotonda. 50 coches y  $t_{max}=2$ .

```
totalCars = 35;
tmax = 2;
```

	1	2	3	4	5	6	7	8	9
1	0	0	2	3	0	0	1	240	0
2	0	0	3	4	0	0	1	260	0
3	0	0	2	3	0	0	1	240	0
4	0	0	1	4	0	0	1	140	0
5	0	0	4	3	0	0	1	200	0
6	0	0	1	4	0	0	1	140	0
7	0	0	4	3	0	0	1	200	0
8	0	0	2	4	0	0	1	240	0
9	0	0	2	3	0	0	1	240	0
10	0	0	4	3	0	0	1	200	0
11	0	0	2	4	0	0	1	240	0
12	0	0	4	2	0	0	1	280	0
13	0	0	4	3	0	0	1	200	0
14	0	0	3	4	0	0	1	260	0
15	0	0	2	3	0	0	1	240	0
16	0	0	4	1	0	0	1	280	0
17	0	0	4	1	0	0	1	280	0
18	0	0	2	1	0	0	1	160	0
19	0	0	1	2	0	0	1	220	0
20	0	0	2	1	0	0	1	160	0
21	0	0	3	1	0	0	1	260	0
22	0	0	2	3	0	0	1	240	0
23	0	0	4	1	0	0	1	280	0
24	0	0	3	4	0	1	1	260	0
25	0	0	1	2	0	1	1	220	0
26	0	0	1	3	0	1	1	220	0
27	0	0	3	1	0	1	1	260	0
28	0	0	4	2	0	1	1	280	0
29	0	0	1	2	0	1	1	220	0
30	0	0	3	4	0	1	1	260	0
31	0	0	2	4	0	1	1	240	0
32	0	0	4	2	0	1	1	280	0
33	0	0	3	2	0	1	1	180	0
34	0	0	4	3	0	1	1	200	0
35	0	0	3	4	0	1	1	260	0

Figura 35. Matriz cars. 35 coches.

Con la intención de saturar la rotonda, se ha configurado el tiempo máximo en 2. Como se puede ver en la Figura 35 todos los coches intentarán aparecer en el mismo instante de tiempo,  $t=1$ . Avancemos hasta  $t=15$ . En la Figura 36 podemos

ver los 35 coches acercándose por las calzadas laterales hacia el anillo de la rotonda. En la Figura 32 los primeros coches han accedido a la calzada anular.

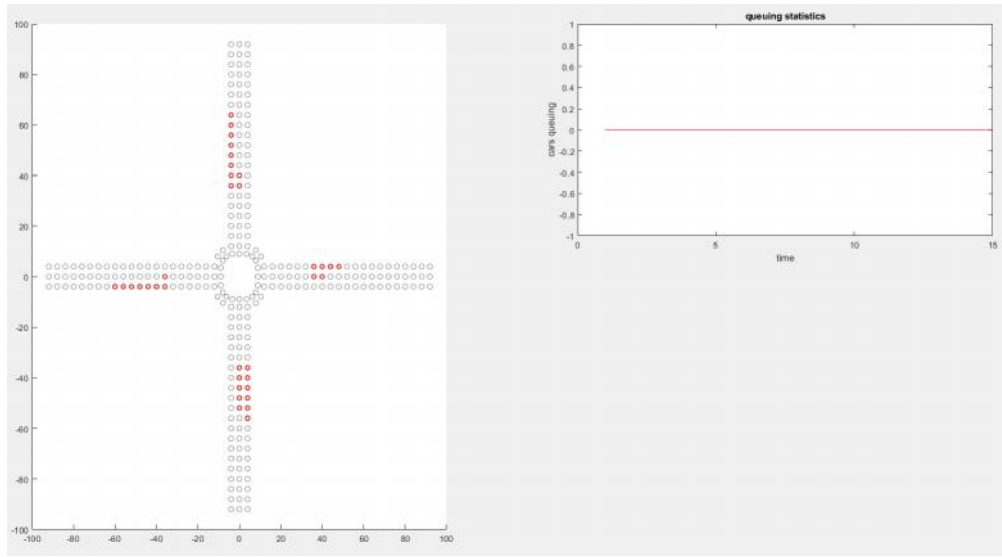


Figura 36.  $t=15$ , 35 coches.

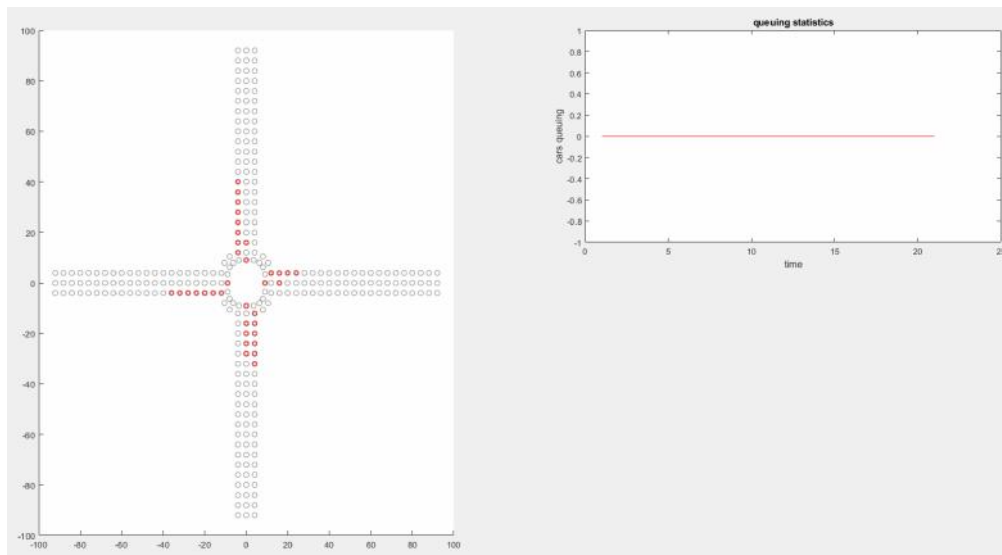


Figura 37.  $t=22$ , 35 coches.

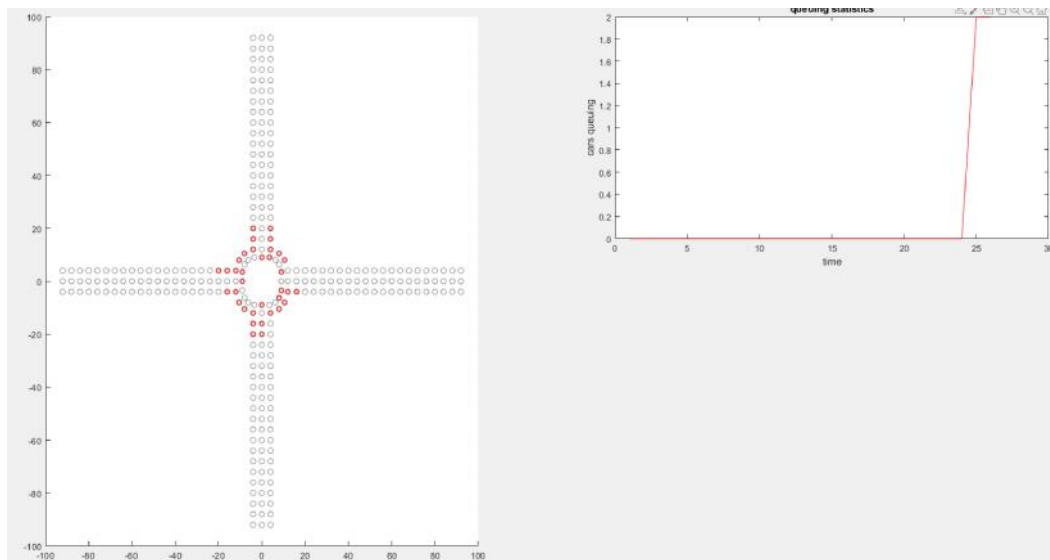


Figura 38.  $t=25$ . coches en espera

En  $t = 25$  vemos como surgen los primeros conflictos en el movimiento de los coches, y aparecen algunos coches en espera. En este caso serán los vehículos que están en la calzada de acceso Sur, en el carril izquierdo, los que están en espera. Como vemos en el gráfico de la estadística de cola podemos ver que coincide y tenemos 2 coches en cola. Al haber coches circulando por el carril externo del anillo los coches del carril de entrada Sur se mantienen en espera, ante la posibilidad de que algún coche continuara su camino por el carril externo en lugar de salir por la salida Sur. Esta situación no sería tan exagerada, ya que los coches que tengan intención de tomar la salida deberían indicar la maniobra mediante el uso del intermitente. Aunque sabiendo cómo funciona la gente a la hora de poner intermitentes, quizás esta situación no sea tan exagerada.

El caso es que, en esta situación, el simulador está pensado para que los coches se mantengan a la espera si ven algún coche acercándose, algo que parece lógico. De forma que estos dos vehículos se mantendrán en cola hasta aproximadamente  $t=37$ , Figura 39, cuando se den las condiciones de seguridad para poder incorporarse al carril interior de la rotonda. Durante el proceso otros vehículos se han añadido a la cola y se han quitado de ella. Durante el paso de

estos 35 vehículos ha habido un máximo de 5 coches en espera durante un instante, para luego irse reduciendo.

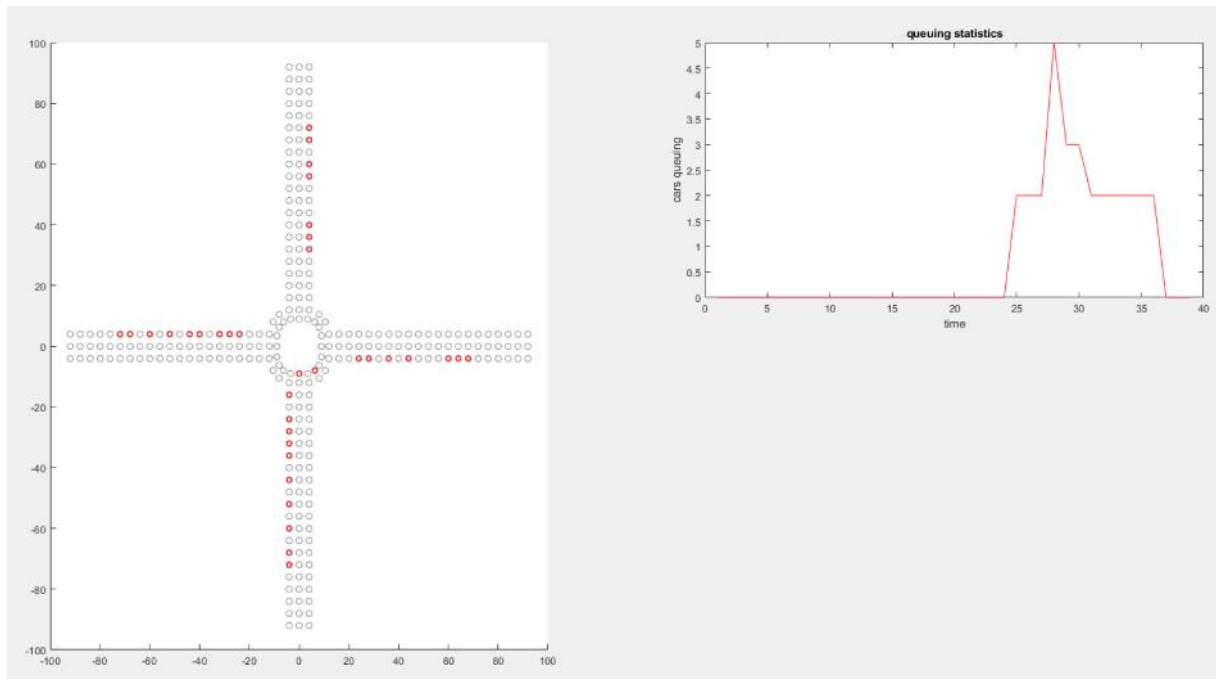


Figura 39. Vaciado de la rotonda

Otra prueba, esta vez con un elevado porcentaje del tráfico que proviene de 2 vías, 50% desde el Este y 30% desde el Norte. Un 80% de los coches tendrán de destino el Oeste. El número de coches configurado son 400 en  $t_{max}=360$ . Esto resulta en una vía con un tráfico moderado y ligero en las demás. Veamos cómo afecta esto al tráfico de la rotonda. El resultado en la Figura 40. La configuración es la que sigue:

```
freq = ...
[50 10 80 10;...
 30 80 10 10;...
 10 20 40 40;...
 10 10 10 80];
```

```
totalCars = 400;
tmax = 360;
```



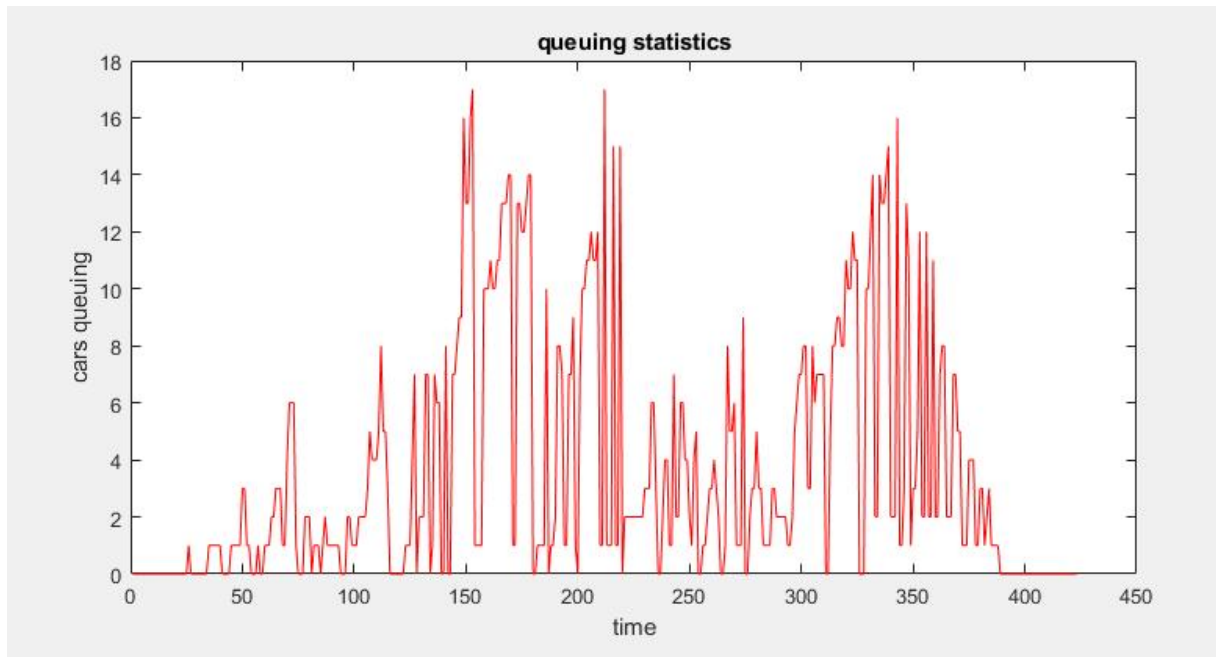


Figura 40. Estado de la cola de coches.

Como es de esperar el elevado tráfico de la vía Este entorpece el paso de coches en la vía Norte, resultando en retenciones más o menos constantes a lo largo del proceso. Esta situación suele ser bastante habitual, una rotonda en la que convergen 4 vías, 1 o dos con una carga de tráfico más elevada y la mayoría de los vehículos que toman la misma salida. Vamos a ver esta misma situación en “hora punta” con la vía Este totalmente saturada. Para ello duplicamos la cantidad de coches, manteniendo el tiempo total. Esto resulta en una rápida saturación de la rotonda. En la Figura 42 se puede ver como ya en  $t=50$ , el número de coches en espera aumenta drásticamente, creciendo sin parar a medida que llegan más coches a la rotonda. Esta situación, aunque parezca exagerada no lo es tanto, es habitual que varias vías se junten con una vía principal y que los vehículos busquen tomar una misma salida. Esto junto a una situación problemática como puede ser el fin de la jornada laboral, o su

comienzo, salida de vacaciones o regreso de estas, puede provocar situaciones parecidas que seguro todos recordamos.

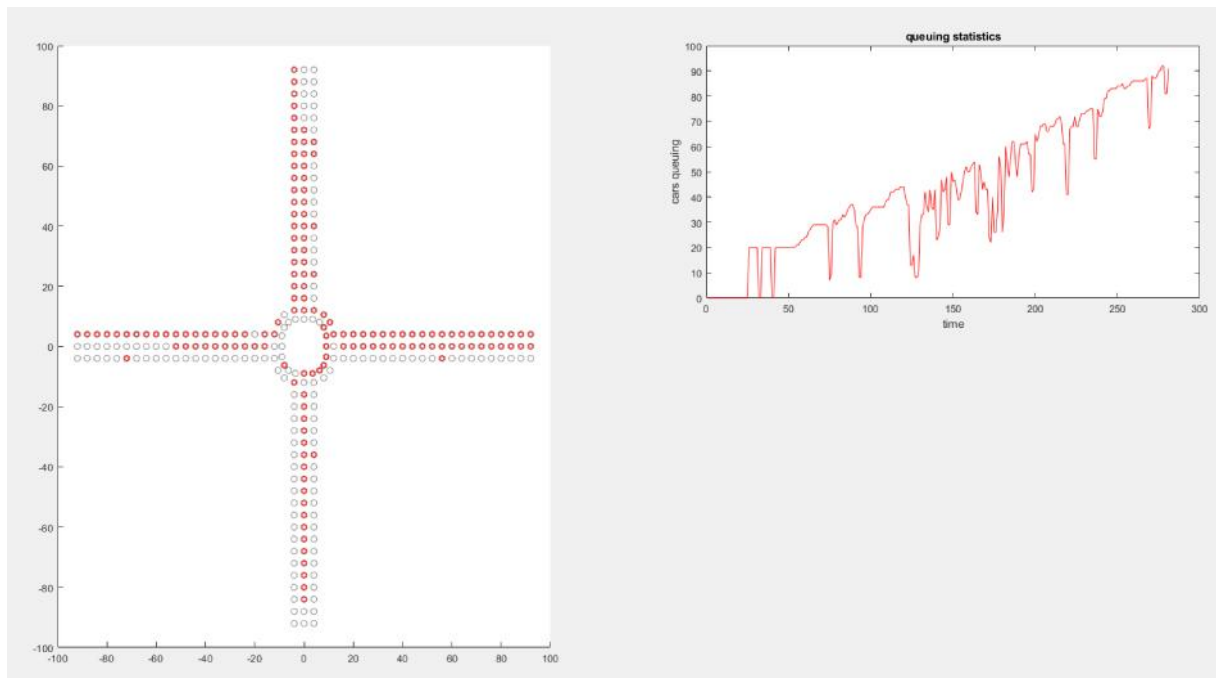


Figura 41. Rotonda saturada, carga elevada

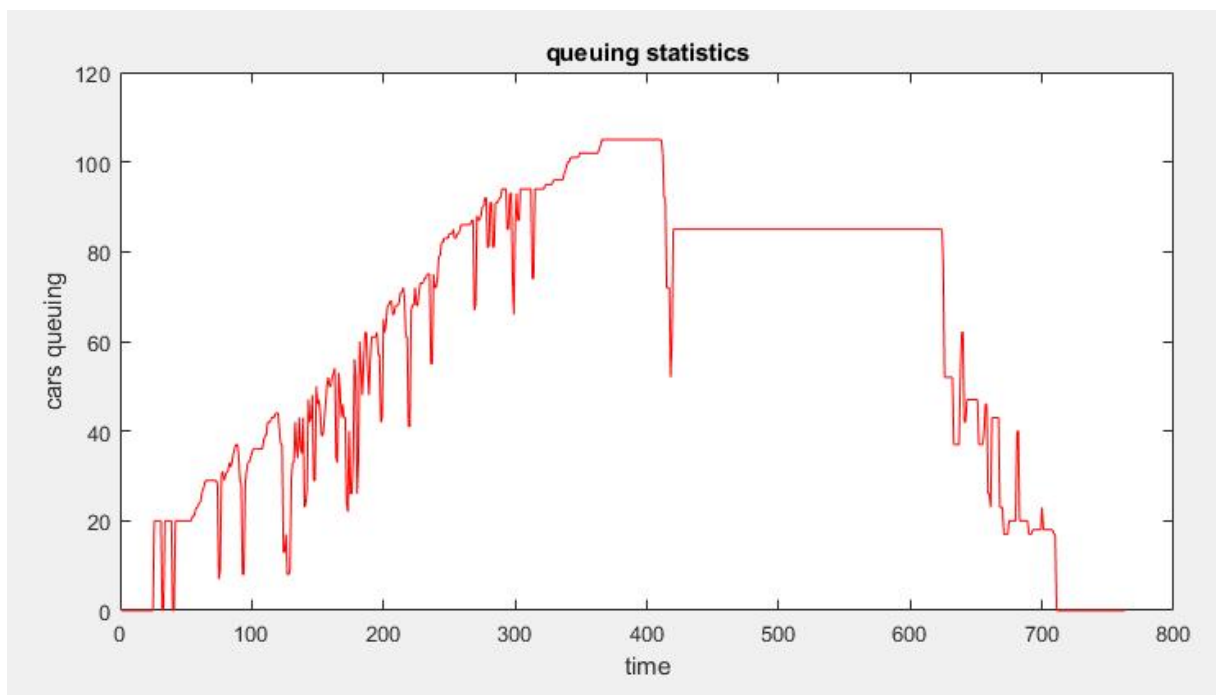


Figura 42. Resultado final.

En la Figura 42 vemos como el número de coches en espera ha crecido sin parar hasta llegar a  $t=360$ , momento en que dejan de llegar coches a través de las calzadas laterales. A partir de ahí el tráfico empieza a reducirse. Se puede ver cómo entre  $t=400$  y  $t=600$  hay un aparente congelamiento del estado de la cola. Esto se debe a la programación de la dinámica de movimientos en la rotonda.

Los carriles laterales de las vías de acceso y el carril exterior de la calzada anular son los primeros en liberarse de tráfico, ya que los vehículos que circulan por la calzada anular exterior son los que más accesibilidad tienen a las salidas. Pero la saturación de la calzada interna del anillo provoca que los carriles de acceso izquierdos de las calzadas laterales queden congelados. Debido a que los vehículos que están en estas calzadas buscan acceder al carril interno, en ningún momento se cambian a los carriles derechos para pasar por el carril externo. Esta opción no está contemplada en la programación. Con esto, cabe pensar que mediante un programa más “realista” o mejor programado, a partir de  $t=360$  cuando dejan de llegar vehículos a la rotonda, la rotonda debería vaciarse rápidamente, ya que los coches utilizarían ambos carriles para llegar a su destino.

### 4.3 Desarrollo del Simulador

A continuación, vamos a utilizar esta representación gráfica, así como la dinámica de movimiento para buscar soluciones a los problemas de tráfico en las rotondas. Y ver su impacto sobre al flujo de tráfico a través de la intersección. Las opciones que se van a evaluar son:

- Añadir carriles laterales a la rotonda, de forma que los vehículos que pretendan coger la primera salida puedan evitar pasar a través del anillo central.
- Usar semáforos inteligentes, que en función del tráfico de cada calle se abran o se cierren.
- Usar un semáforo “giratorio”. La idea es un círculo con un sector verde que gira en sentido horario por encima de la rotonda. Serán los vehículos que queden dentro del sector verde los que puedan continuar la marcha y acceder a la rotonda.
- Utilizar un semáforo giratorio con 4 sectores. En el que serán los colores correspondientes a cada entrada los que permitan el paso a la rotonda.

Para representar los vehículos, vamos a usar triángulos en lugar de círculos. Los triángulos estarán rellenos del color que identifique desde donde vienen. Además, apuntarán hacia donde se dirigen. De forma que un triángulo que apunte hacia arriba irá al Norte y hacia abajo al Sur etc. Los colores son: verde-Este, azul-Norte, rojo-Oeste, magenta-Sur.

Para poder personalizar la ejecución de nuestro simulador, creamos una interfaz, Figura 43. En ella tenemos dos zonas, en la zona superior tenemos la

conFiguración de la simulación y en la zona inferior la visualización de los resultados.

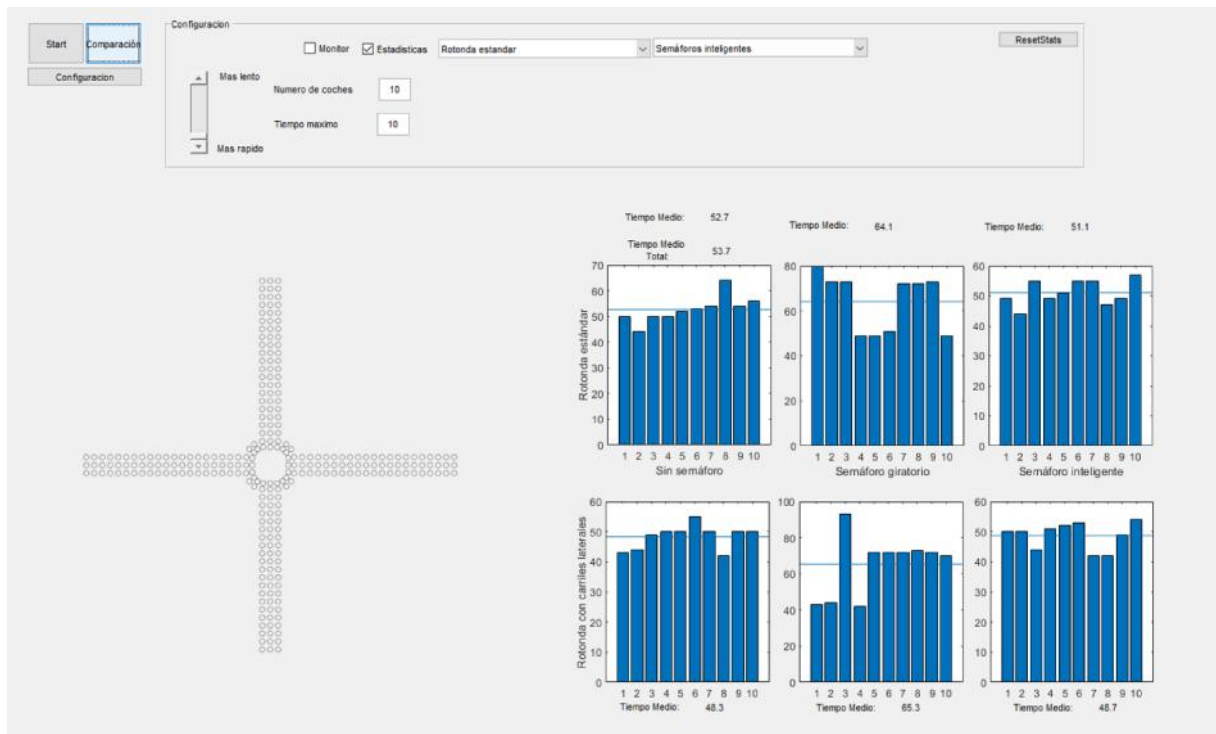


Figura 43. Interfaz

Dentro de la configuración tenemos:

- Start: inicia la simulación.
- Configuración: oculta el cajón de configuración.
- Comparación: ejecuta las 6 posibles combinaciones de rotondas. Rotonda estándar sin semáforo, rotonda estándar con semáforo giratorio, rotonda estándar con semáforo inteligente, rotonda con carriles laterales sin semáforo, rotonda con carriles laterales con semáforo giratorio, rotonda con carriles laterales con semáforo inteligente.
- Slider más lento- más rápido: permite aumentar o reducir la velocidad de la simulación.
- Checkbox monitor: Muestra u oculta la representación gráfica de la simulación. Esta representación se puede ver abajo a la izquierda. El desactivar esta opción acelera enormemente la velocidad de ejecución de la simulación.
- Número de coches: número total de coches que pasarán por la rotonda.
- Tiempo máximo: tiempo máximo para que los coches entren en la rotonda. Cabe señalar que por cada carril solo podrá circular un único coche a la vez, de forma que, si dos coches deben entrar al mismo tiempo por un carril, primero entrará uno de ellos y después el siguiente.
- Listbox selección rotonda. Permite cambiar entre la rotonda estándar, y la rotonda con carriles laterales.
- Listbox selección señalización rotonda. Selecciona entre los distintos tipos de señalización disponibles.

- Botón ResetStats: Reinicia el valor del tiempo medio total guardado hasta el momento.

En la zona de visualización:

- A la izquierda se representa la rotonda seleccionada en caso de que la opción monitor esté seleccionada
- A la derecha se representan las estadísticas. Los gráficos serán de barras, cada barra representa el tiempo total que ha necesitado cada coche para llegar a su destino. También se muestra una línea de referencia que señala el tiempo medio del grupo.

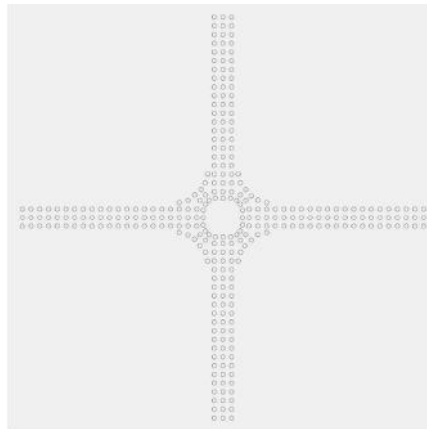


Figura 44. Rotonda con carriles laterales

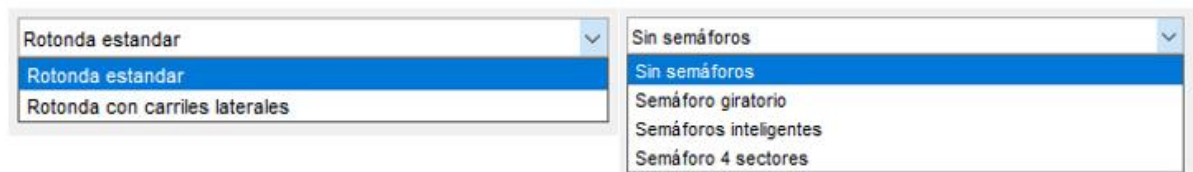


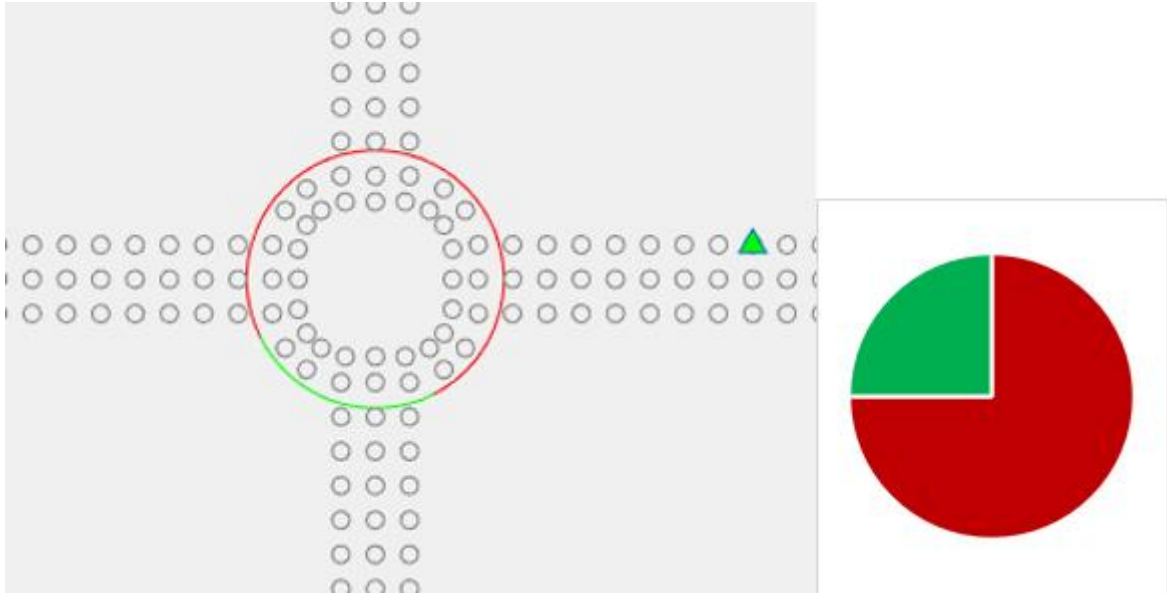
Figura 45. Listbox selección parámetros

Vamos a ver con más detalle las funcionalidades del simulador. Podemos seleccionar rotonda estándar o rotonda con carriles laterales. La diferencia está en que, la rotonda con carriles laterales tiene carriles adicionales, que permiten que los vehículos con intención de tomar la primera salida puedan evitar circular a través de la rotonda. En la Figura 44 se puede ver un ejemplo. Al seleccionar una u otra configuración, se modifican distintos parámetros, como las variables rotonda, semáforo, que se encargan de seleccionar el tipo de rotonda, así como el tipo de señalización que tendrá la rotonda. El código encargado de esta parte lo vamos a comentar más adelante, en la sección correspondiente al botón comparación.

En la otra listbox podemos escoger entre varias opciones de señalización:

- Sin semáforos sería una rotonda con preferencia a la izquierda clásica. En la que son los vehículos que circulan por el anillo los que tienen prioridad, mientras que los vehículos que se aproximen a la rotonda cederán el paso a los coches que se aproximen por su izquierda.

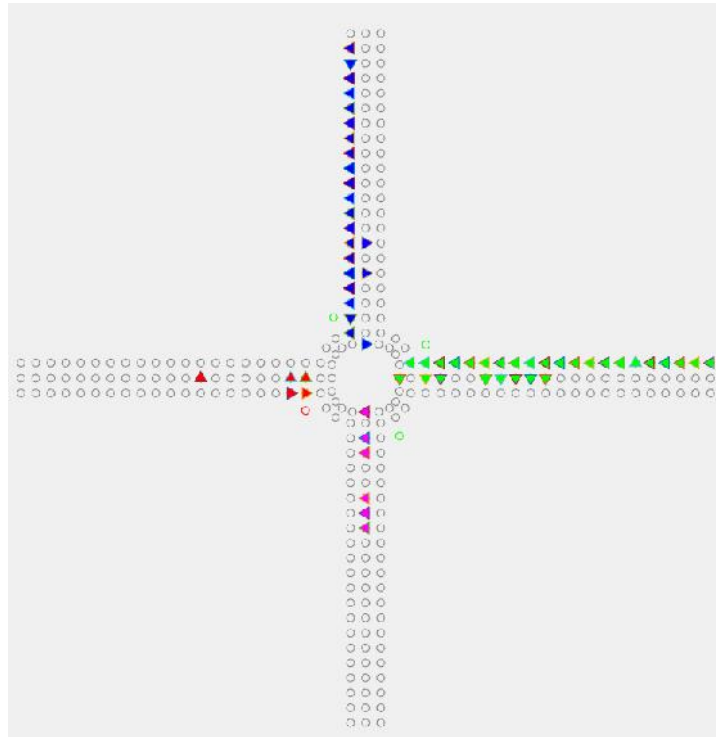
- Semáforo giratorio. La idea es la siguiente, un área con un sector rojo y una parte verde, como se puede ver en la Figura 46. Serán los coches del área verde los que tengan permiso de paso a la calzada anular. Para evitar que el área de permiso de paso acompañe a los vehículos que circulan por la rotonda, y se evite así el paso de nuevos vehículos, el área circular girará en el sentido contrario a la circulación, esto es, horario.



*Figura 46. Concepto semáforo giratorio.*

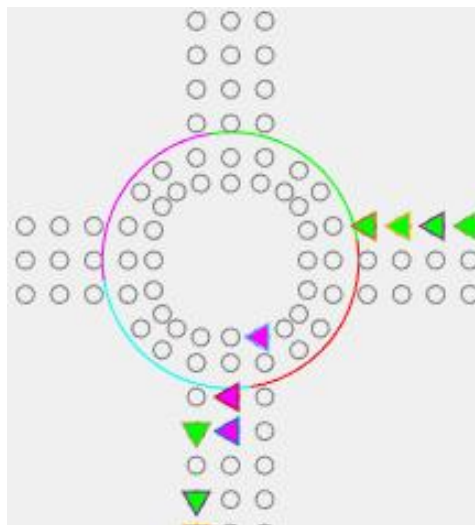
- Semáforos inteligentes: como alternativa a los semáforos clásicos, regidos por intervalos de tiempo. Estos semáforos “inteligentes” cambiarían de forma que sean las vías con menor carga de tráfico las que tengan que esperar, con el fin de que pasen el mayor número de coches por la rotonda lo más rápido posible. Están representados con pequeños círculos de color rojo o verde, dependiendo de su estado. Ejemplo en la

Figura 47, el semáforo Oeste retiene los coches para favorecer el paso en las vías con más carga.



*Figura 47. Semáforos inteligentes*

- Semáforo 4 sectores: esta opción (Figura 48). En este tendríamos 4 sectores identificados con los colores de las vías. Cada sector de un color permitirá el paso a la vía de su color. Esta da lugar a diversas posibles modificaciones, se podría cambiar el orden de los sectores, su anchura, la velocidad de giro y ver qué resultados se obtiene.



*Figura 48. Semáforo 4 sectores*



- Semáforo 4 sectores dinámicos: Una variación del semáforo de 4 sectores. En este semáforo existirían 2 sectores que reducirán su tamaño para dar paso a los otros 2 sectores. Esta idea es un concepto, que no se ha llegado a implementar.

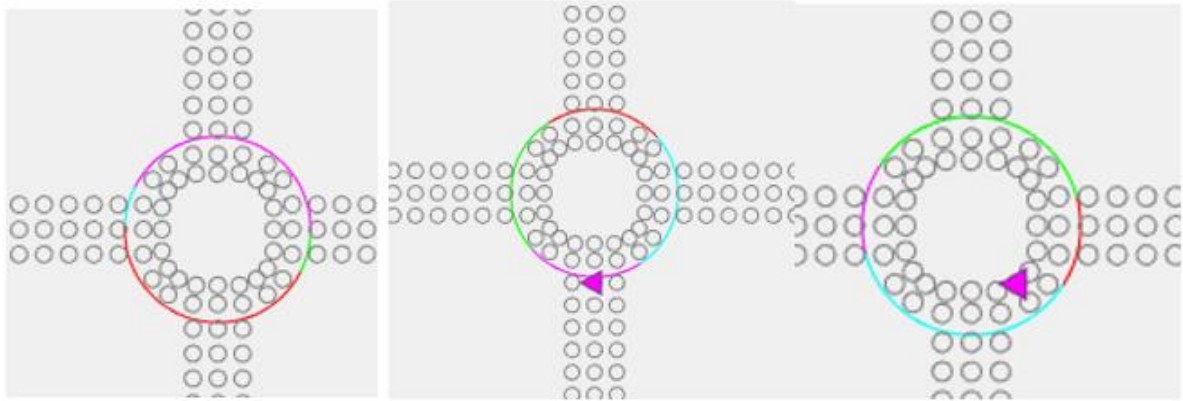


Figura 49. Semáforo 4 sectores dinámicos

Existen algunas opciones de configuración que no se reflejan en la interfaz. De la matriz **freq** ya hemos hablado anteriormente. La configuración por defecto sería similar a la que hemos usado antes:

```
freq = ...
[50 10 80 10;...
 20 80 10 10;...
 15 20 40 40;...
 15 10 10 80];
```

La mayor parte del tráfico llegará por la vía 1 (Este), por las demás la carga será bastante más reducida, 20% Norte, 5% Sur y 5% Oeste. El tráfico de las dos vías con más carga circulara en su mayoría (80%) hacia el Oeste. El objetivo de esta configuración es simular una rotonda en la que, una de sus calles posee una carga de tráfico considerablemente superior y en la que se daría lugar a atascos debido a problemas de autobloqueo en la rotonda.

Se ha incluido en las funciones que controlan el movimiento de los coches, una nueva característica. Con objetivo de simular el error humano a la hora de realizar una incorporación a la calzada central, o al salir de ella, hay una nueva condición que se debe cumplir. Mediante la variable **prob** (por defecto 5 [%]), además de requerirse que la celda a la que el coche se moverá esté libre, debe darse que, la generación de un numero aleatorio entre 0 y 100 resulte en un valor superior al valor de **prob** para que el coche pueda avanzar. Esto simula una posible distracción a la hora de incorporarte a un carril, lo que resultaría en la perdida de la oportunidad y por tanto tener que esperar a disponer del espacio para poder avanzar.

También se debe dar que, una vez liberada la celda a la que el coche se moverá, deberá quedar libre a su vez la siguiente casilla. Esto simula el proceso de

aceleración de un coche, y el tiempo que tarda en llegar a su nueva posición dentro del carril.

- Comparación: este botón llama una rutina especial. Mediante un bucle, se realizan 6 simulaciones consecutivas. En todas ellas se utilizará el número de coches y el **Tmax** indicado en la configuración. Se configura automáticamente monitor=0 para evitar que se muestre el proceso, y así agilizar el procesamiento. Las variables que controlan la selección del tipo de rotonda y de la señalización son rotonda y semáforo. En cada iteración del bucle se seleccionará una de ellas, primero la rotonda estándar sin semáforos, rotonda estándar con semáforo giratorio y la rotonda estándar con semáforo inteligente. Después la rotonda con carriles laterales con las mismas configuraciones en el mismo orden.

Esto está dentro de la función Callback del boton6, el botón comparación. A continuación, se muestra el código correspondiente a esta parte y la parte encargada de generar las estadísticas.

```
ncochesstring = findobj('Tag', 'edit2');    %número de coches recogido
- edit2
ncoches=str2double(get(ncochesstring,'string'));
tmaxstring = findobj('Tag', 'edit4');    %tiempo maximo - edit4
Tmax=str2double(get(tmaxstring,'string'));
monitor=0;    %enseñar o no el movimiento de los coches en pantalla
pausa=0;
svvalue=25;
for it=1:1:6
    switch it
        case 1
            semáforo=1;
            rotonda=1;
        case 2
            semáforo=2;
            rotonda=1;
        case 3
            semáforo=3;
            rotonda=1;
        case 4
            semáforo=1;
            rotonda=2;
        case 5
            semáforo=2;
            rotonda=2;
        case 6
            semáforo=3;
            rotonda=2;
    end
cars=Main(ncoches,monitor,semaforo,rotonda,Tmax,pausa, svvalue);

bucle=handles.bucle;    %bucle controla el array para las estadísticas
en caso de ejecutar varias veces seguidas
bucle=bucle+1;
handles.bucle=bucle;
stats_show=1;    %enseñar o no las estadísticas
media= sum((cars(:,9)-cars(:,6))/(length(cars(:,1)))));    %tiempo medio
que tardan los coches en pasar por la rotonda
```

```

medias(bucle)=media; %media total en caso de ejecutar
el programa varias veces seguidas
handles.medias(bucle)=medias(bucle);

set(handles.text8, 'String',
sum(handles.medias)/length(handles.medias));
if(stats_show)

    set(handles.uipanel1,'visible','on');
    set(handles.uipanel2,'visible','on');
    switch it
    case 1
        axesHandle = findobj('Tag', 'axes2');
        bar(axesHandle,(cars(:,9)-cars(:,6))) %grafico de barras
        media= sum((cars(:,9)-cars(:,6))/(length(cars(:,1)))));
        refline(axesHandle,0,media);
        set(axesHandle,'Tag', 'axes2') %renombrar el tag
de axes2 para volver a buscarlo en la sgte iteracion
        set(handles.text3, 'String', media);
        xlabel(axesHandle,'Sin semáforo')
        ylabel(axesHandle,'Rotonda estándar')

```

Una vez se han cargado los parámetros de la simulación, se envían a la función **Main.m** que se encarga de toda la simulación, vamos a verla más adelante. **Main.m** devuelve la matriz cars, que contiene toda la información de la simulación como ya hemos visto. De esta forma a partir de cars podemos obtener la información de cómo ha ido la simulación. Se calcula la media, se resta al tiempo de llegada al fin de trayecto de cada coche el tiempo de inicio, se suman todos y se divide entre el número de coches:  $media = \frac{\sum (cars(:,9) - cars(:,6))}{length(cars(:,1))}$  Para representar la media se dibuja una línea horizontal de referencia sobre el gráfico de barras. El gráfico de barras se

obtiene con el tiempo total de trayecto de cada coche, es decir el tiempo de fin menos el tiempo de inicio.

Este proceso se repite para las distintas combinaciones de rotonda y semáforo. Finalmente se representan en una cuadrícula 2x3. El resultado se muestra en la Figura 50.

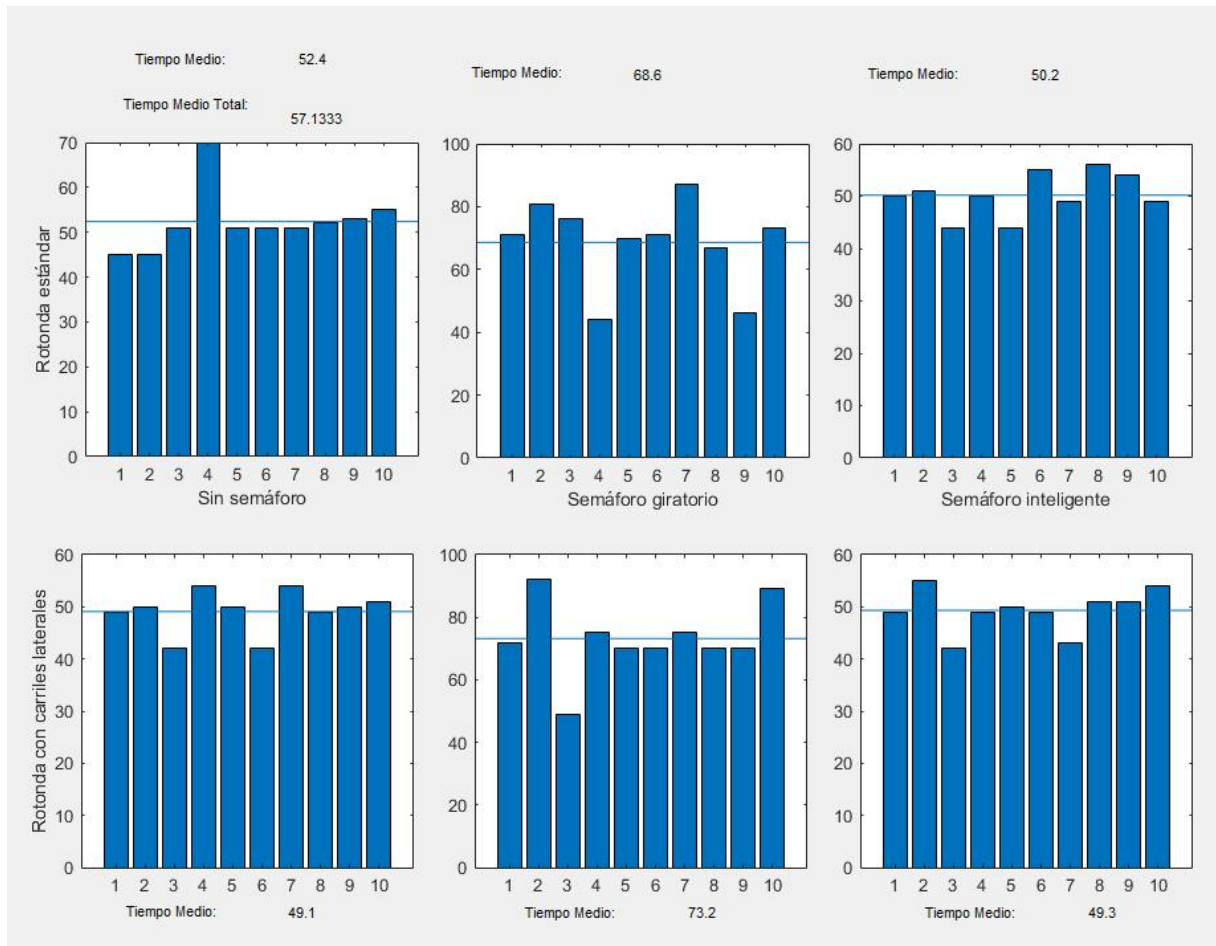


Figura 50. Gráficos de barras

Ahora vamos a ver la función **Main.m**, encargada del manejo de las funciones de inicialización de la simulación: **initCars.m**, **initGeom.m** y **initGeomA.m**. Esta última función es una adaptación de **initGeom.m** para utilizarla con la rotonda que incluye carriles laterales. De forma similar fue necesario adaptar **moveCarToCross.m** tanto para su utilización en la rotonda con semáforos, como para su uso en la rotonda con carriles laterales, estas funciones son: **moveCarToCrossSemaforo.m** y **moveCarToCrossSemaforoA.m**. La diferencia fundamental es que en las condiciones necesarias para que un vehículo pase de la última celda de un carril de entrada pase a la calzada anular se añade la condición de que el semáforo se encuentre en verde. En el caso de la rotonda con carriles laterales, se incluye una ruta alternativa, que va desde el carril de entrada derecho hacia el carril de salida de la siguiente vía. Todas las funciones están incluidas en el Anexo 3. La función **Main.c**:

```
function [cars]=Main(ncoches,monitor,semaforo,rotonda,tiempo,pausa,svvalue)
```

```

tmax=tiempo;
% escenario 1
% freq = ...
%     [35 70 15 15;...
%       35 15 15 70;...
%       15 20 40 40;...
%       15 40 40 20];
% escenario 2

freq = ...
    [60 10 80 10;...
      20 80 10 10;...
      10 20 40 40;...
      10 10 10 80];

totalCars = ncoches; % totalCars = 7500;
coches=zeros(totalCars,7); %#ok<PREALL>

strategy = [0;100];
destino=['>', '^', '<', 'v'];
color=['g', 'b', 'r', 'm']; %Verde, azul, rojo, morado
show = 1;

```

Aquí vemos dos posibles escenarios que dependen del valor de la matriz freq. El valor de freq se puede modificar en el código de la función **main**, ya que su modificación a través de la interfaz no está implementada.

```

%% Inicializar semáforo
tiposemaforo=semáforo;
sem=[4,3,2,1];
s=[1,2,3,4];
semaforo_cont=1;
sectorverde=svvalue/100;
switch tiposemaforo
    case 2
        g=2*pi-pi/4-5*pi/30:-pi/30:-pi/4-5*pi/30;
        v=2*pi*sectorverde:-pi/30:0;
        r=2*pi:-pi/30:2*pi*sectorverde;
        x=15*cos(v);
        y=15*sin(v);
        X=15*cos(r);
        Y=15*sin(r);
        [T,R]=cart2pol(X,Y);
        [t,r]=cart2pol(x,y);

    case 3
        lucesx=[16.0561,-8.5554,-16.0561, 9.0242,];
        lucesy=[8.9070,16.1732, -8.6726, -15.4701];
        [tsem,rsem]=cart2pol(lucesx,lucesy);

    case 4
        g=2*pi-pi/4-5*pi/30:-pi/30:-pi/4-5*pi/30;
        v=2*pi:-pi/30:1.5*pi;
        a=0.5*pi:-pi/30:0;
        m=pi:-pi/30:0.5*pi;
        r=1.5*pi:-pi/30:pi;
        vx=15*cos(v);
        vy=15*sin(v);

```

```

        rx=15*cos(r);
        ry=15*sin(r);
        ax=15*cos(a);
        ay=15*sin(a);
        mx=15*cos(m);
        my=15*sin(m);
        [tr,rr]=cart2pol(rx,ry);
        [tv,rv]=cart2pol(vx,vy);
        [ta,ra]=cart2pol(ax,ay);
        [tm,rm]=cart2pol(mx,my);
end

```

Según los parámetros enviados a la función **Main.c** se elegirá la inicialización correcta para el semáforo. Siendo los valores de la variable **tiposemaforo**:

- 1 sin semáforos
- 2 semáforo giratorio
- 3 semáforo inteligente
- 4 es la parte correspondiente al semáforo de 4 sectores, la cual está inacabada. Como vemos la representación gráfica del semáforo está lista. Los vectores **r,a,m,v** contienen los radianes que abarcará cada sector (rojo, azul, magenta y verde respectivamente). Esta información se pasa a las variables **tr,rr,tv,rv** para su representación polar más adelante.

```

%% Initialize geometry and cars
switch rotonda
    case 1
        cases = initGeom(show);
    case 2
        cases = initGeomA(show);
end
cars = initCars( freq, totalCars, tmax, strategy);
%% initialize statistical variables
finalt=0;
coches=[];

```

Aquí se inicializa la geometría de la rotonda, en función del parámetro **rotonda**, y la matriz **cars**, con los parámetros **freq**, **totalCars**, **tmax** y **strategy**. Ahora vamos a ver el bucle principal de la función **Main.m**. El código completo está en el anexo 3, aquí, para facilitar su comprensión vamos a ir comentándolo por secciones:

El fin de la ejecución está marcado por la variable **totalCars**. Cuando el número de coches que han llegado a su destino (columna 7 de **cars** con valor 2: **cars(:,7)==2**) se saldrá del bucle.

```

%% Time loop
while(length(find(cars(:,7)==2))~=totalCars)
    %% loop until all cars have reached their destination
    finalt=finalt+1;

    %% Semaforo
switch tiposemaforo

```

```
% Sin Semáforo
case 1
    sem=[1,2,3,4];
    s=[1,2,3,4];
    sem4=zeros(1,4);
```

El primer caso es la rotonda sin semáforos. Los valores de **sem** , **s** y **sem4** (**sem4** es el array pensado para ser utilizado con el semáforo de 4 sectores) son los que tendríamos en caso de que todos los semáforos estuvieran en verde. De forma que, de cara a la simulación, la rotonda sí tiene semáforos, pero estos estarán siempre en verde.

```
case 2
    sem4=zeros(1,4);
    semaforo_cont=semaforo_cont+1;
    if semaforo_cont==61
        semaforo_cont=1;
    end

    s=ones(1,4)*ceil(semaforo_cont/15);
    polarplot(t+(g(semaforo_cont)),r,'g');
    polarplot(T+g(semaforo_cont),R,'r');
    axis off
```

**tiposemaforo==2** se corresponde con el semáforo giratorio. La variable **semáforo\_cont** se incrementa cada vez que el semáforo giratorio gira un paso (al llegar al valor final 61 vuelve a valer 1). Con esta variable **semáforo\_cont** podemos dividirla entre 15 para saber qué porcentaje de vuelta ha completado y de esta forma abrir y cerrar el paso a través de las distintas vías. 15 es el 25% de 60, que son los pasos que tiene que dar el semáforo para dar una vuelta completa. De forma que si quisiéramos aumentar o disminuir la zona verde habría que modificar el valor 15 para adaptarlo a la nueva situación.

En el array **s** se guardan los valores que indicarán qué vías tendrán el paso abierto a la rotonda. El array **s** será siempre un vector con 4 números iguales. Como hemos comentado antes los valores de **s** que abren los semáforos son **sem=[1,2,3,4]** , de forma que únicamente se dará paso a una vía cada vez, [1,1,1,1] abrirá la vía Este. [2,2,2,2] abrirá la vía Norte etc. El valor de **s** se obtiene al dividir **semáforo\_cont** entre 15 (redondeando hacia arriba), lo que nos dará un 1 cuando el sector verde esté mirando al Este, por lo que **s** será [1,1,1,1], [2,2,2,2] cuando el sector verde ocupe el Sur. Para que coincida **s** con los números que identifican las vías, recordemos que están numeradas en sentido antihorario, 1-Este y 4-Sur. Se realiza una transformación con el array **sem(4,3,2,1)** de la siguiente forma: **sem(s(2))** se utiliza para verificar si la segunda vía, el Norte, tiene permiso de paso. De forma que si **sem(s(2))==2** el Norte puede continuar circulando. Esto será cierto cuando el semáforo giratorio haya recorrido más de 45 pasos, y por lo tanto el vector **s** sea [3,3,3,3]. **s(2)** contiene un 3, luego **sem(3)** es igual a 2 y el Norte tiene permiso de continuar.

```
case 3
    sem4=zeros(1,4);
    cc(1)=length(vertcats(find(cases(121:140,1)
0),find(cases(201:220,1) > 0))); %coches calle 1 ...
    cc(2)=length(vertcats(find(cases(141:160,1)
0),find(cases(221:240,1) > 0)));
    cc(3)=length(vertcats(find(cases(161:180,1)
0),find(cases(241:261,1) > 0)));
```



```

        cc(4)=length(vertcat(find(cases(181:200,1)
0),find(cases(261:280,1) > 0))); %
        minimo=min(cc);

        for a=1:1:4
            if (cc(a)== minimo || cc(a)<minimo)
                sem(a)=5;
            else
                sem(a)=a;
            end
        end
        if (length(unique(sem))==1) %comprobar si están todos los
semáforos en rojo
            sem=[1,2,3,4];
        end
end

```

El semáforo inteligente funciona de la siguiente manera. Hace un recuento de los coches que hay en cada vía de acceso. Y a aquella vía con menos coches le asigna a su posición en el array sem un valor 5. Como hemos visto las vías tienen el semáforo en verde cuando en sem coincide que en la posición correspondiente a la vía se encuentra el id de esa vía. Por tanto, al asignar un 5 hacemos que el semáforo esté en rojo. A las demás vías se les asigna un valor igual a su posición, para asegurar que los semáforos están en verde. En caso de que todas las vías tengan el mismo número de coches (todos los semáforos se pondrían en rojo) nos aseguramos de que los semáforos estén en verde para que los coches puedan circular `sem=[1,2,3,4]`

```

case 4
    if semaforo_cont==1
        semaforo_cont=61;
    end
    semaforo_cont=semaforo_cont-1;
    s=ones(1,4)*ceil(semaforo_cont/15);
    polarplot(tv+g(semaforo_cont),rv,'g');
    polarplot(tr+g(semaforo_cont),rr,'r');
    polarplot(ta+g(semaforo_cont),ra,'c');
    polarplot(tm+g(semaforo_cont),rm,'m');
    sline4=[1, 2, 4, 3];
    sem4v=horzcat(ones(1,15)*sline4(3),ones(1,15)*sline4(4),
ones(1,15)*sline4(1),ones(1,15)*sline4(2));
    sem4vp=horzcat(sem4v(semaforo_cont:end),
sem4v(1:semaforo_cont));
    sem4=horzcat(sem4vp(1),sem4vp(15),sem4vp(30),sem4vp(45));
    axis off
    s=[1,2,3,4];
    sem=[1,2,3,4];

end
end

```

La clave del semáforo de 4 sectores está en el array sem4v. Se trata de un array formado por tantos 1 como posiciones abarca el sector verde del semáforo, lo mismo con los 2 para el sector azul, 3 para el sector rojo y 4 para el sector magenta. Este array se rota utilizando la variable semaforo\_cont, según aumenta o disminuye el array sem4v se desplaza y almacena en sem4vp. Desde éste se seleccionan las posiciones correspondientes a las entradas a la rotonda [1, 15,

30, 45] y se comprueba si el número corresponde con el color de la vía, de ser así el semáforo estaría “en verde” y el vehículo puede acceder a la rotonda.

```

        %% Movimiento de los coches
        [cars, cases]=moveCarFromCross(cars, cases, finalt);
        [cars, cases] = moveCarOuterRoundabout(cars, cases);
        [cars, cases] = moveCarInnerRoundabout(cars, cases);
        switch rotonda
            case 1
                [cars, cases]=moveCarToCrossSemaforo(cars, cases,
finalt,s,sem,sem4);
            case 2
                [cars, cases]=moveCarToCrossSemaforoA(cars, cases,
finalt,s,sem,sem4);
            end
        [theta, rho]=cart2pol(cases(:,2),cases(:,3));
        %% Dibujar en la pantalla
        if(monitor)

            polarplot(theta,rho,'o','Color',[0.5 0.5 0.5]);
            axis off
            hold on
            occupied = find(cases(:,1) > 0);
            if(tiposemaforo==3)
                for a=1:1:4
                    if sem(a) == 5
                        polarplot (tsem(a),rsem(a),'o','Color','red');
                    else
                        polarplot (tsem(a),rsem(a),'o','Color','green');
                    end
                end
            end
            if( ~isempty(occupied) )
                if( ~isempty(coches) )
                    delete(coches)
                    clear coches
                end
                for l=1:1:length(occupied)
                    % plot(cases(occupied(l),2),
cases(occupied(l),3),
destino(cars(cases(occupied(l),1),4)), 'MarkerFaceColor',
color(cars(cases(occupied(l),1),3)), 'MarkerSize',8);
                    coches(l)=polarplot(theta(occupied(l)),
rho(occupied(l)),
destino(cars(cases(occupied(l),1),4)), 'MarkerFaceColor',
color(cars(cases(occupied(l),1),3)), 'MarkerSize',8);
                    axis off
                end
            end
            pause(pausa);
        end
    end
    delete(coches)

```

Finalmente está la parte de movimiento de los coches, su representación o no en pantalla, mediante el parámetro monitor, como ya hemos visto anteriormente. Al final del bucle podemos encontrar un pause(pausa), este se encarga de ralentizar la representación gráfica del proceso, permitiendo ajustar la velocidad mediante el slider en la zona de configuración que adjudica un valor a la variable pausa.



## 5. Evaluación

En este capítulo evaluaremos 6 escenarios. En estos escenarios, se proponen modificaciones de la rotonda simple. Buscaremos mejorar cómo circulan los vehículos por la rotonda, especialmente en situaciones complicadas, como puede ser una rotonda donde el tráfico de una de las calles es superior al de las demás, pudiendo dar lugar a saturaciones.

Los escenarios son:

- Rotonda sin semáforos.
- Rotonda con carriles laterales
- Rotonda con semáforo de 4 sectores.
- Semáforo circular con dos sectores intercambiados.
- Semáforo circular con tres colores cambiados. Versión 1.
- Semáforo circular con tres colores cambiados. Versión 2

Para la configuración del tráfico en todos los escenarios vamos a utilizar la matriz **freq** que hemos visto hace un momento, denominada como escenario 2.

```
% escenario 2

freq = ...
    60  10  80  10;...
    20  80  10  10;...
    10  20  40  40;...
    10  10  10  80];
```

Figura 51. Configuración flujo vehículos

Esto significará que nuestra vía principal (Este) tiene un flujo constante de vehículos hacia el Oeste, mientras que las demás tendrán una carga bastante más reducida. A pesar de que el nivel de carga de la vía Norte es bastante reducido, el hecho de que la mayor parte del tráfico se dirige hacia el Oeste provoca que se produzca una aglomeración de vehículos o “atasco”, ya que aquellos que vengan del Norte deberán ceder el paso a todos los demás vehículos que se dirijan hacia el Oeste.

El carácter aleatorio de las simulaciones hace que exista un margen en los resultados. Esto es provocado por factores como la probabilidad de los vehículos de entrar por una vía o por otra, o el factor aleatorio que simula “despistes” de los conductores a la hora de incorporarse a la circulación de la calzada circular.



Figura 52. Configuración.

Por esto se realizarán varias simulaciones y se compararán la media de los resultados.

Dado que las dimensiones de la rotonda no están definidas, ni la velocidad de circulación de los vehículos, el tiempo de la simulación es adimensional. Queda como posible trabajo futuro realizar los cálculos necesarios para asignar las unidades correctas.

### 5.1 Rotonda sin semáforos:

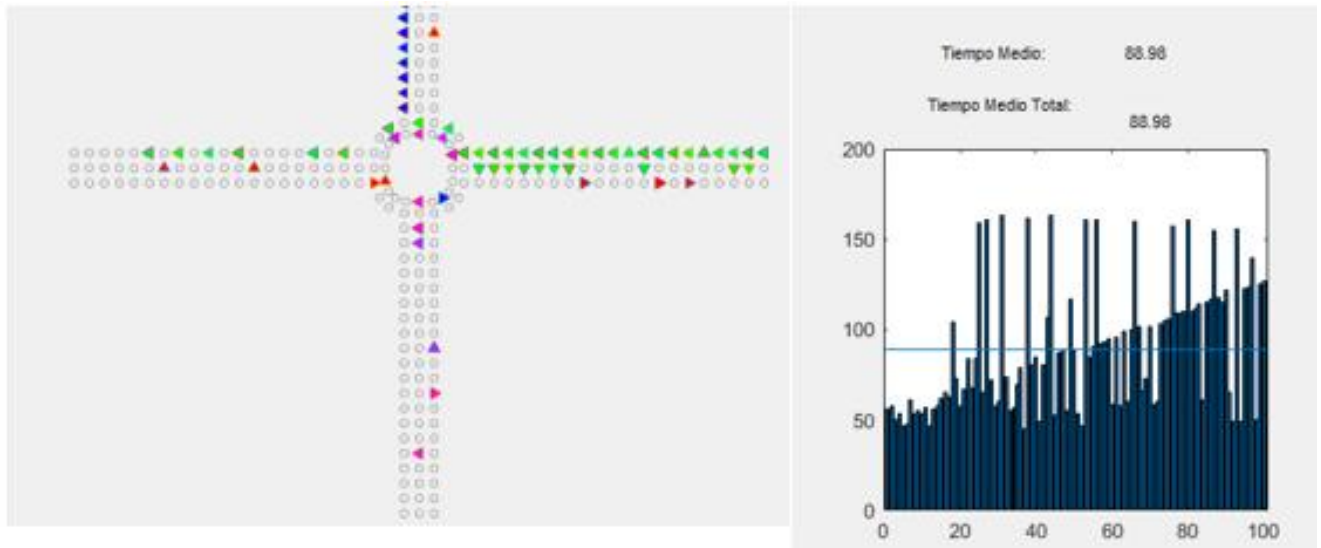


Figura 53. Rotonda estándar.

Con un tiempo medio de trayecto de 89. Se puede ver cómo según aumenta el número de coches en la rotonda, aumenta el tiempo de trayecto de los vehículos. Eso se debe principalmente al autobloqueo de la rotonda y a los atascos que se producen. Además de dichas aglomeraciones, podemos ver en el gráfico cómo hay picos, es decir, coches puntuales que han tenido que esperar un tiempo anormalmente alto para poder atravesar la rotonda. Seguramente fueran coches que llegan por el Norte, obligados a esperar a tener un hueco entre el gran flujo de coches que llega desde el Este.

### 5.2 Rotonda con carriles laterales:

Se trata de una rotonda estándar con posibilidad de evitar el tráfico de la calzada circular si tu intención es tomar la primera salida. Para ello antes de llegar a línea de ceda el paso en el anillo, se deberá tomar una salida a la derecha, que llevará directamente al carril de salida de la siguiente vía. Para incorporarse a este último carril, los vehículos que hayan escogido utilizar el atajo deberán ceder el paso a los vehículos que circulen por la calzada de salida de la rotonda. Frente a la rotonda estándar, esta opción ofrece una mejora de alrededor del 15% en el tiempo de trayecto. Además, como se puede ver en el gráfico, el tiempo no aumenta de forma constante, es decir que la rotonda no se satura, debido a que

la inclusión de estos carriles laterales produce un incremento en la capacidad de la rotonda.



Figura 54. Configuración

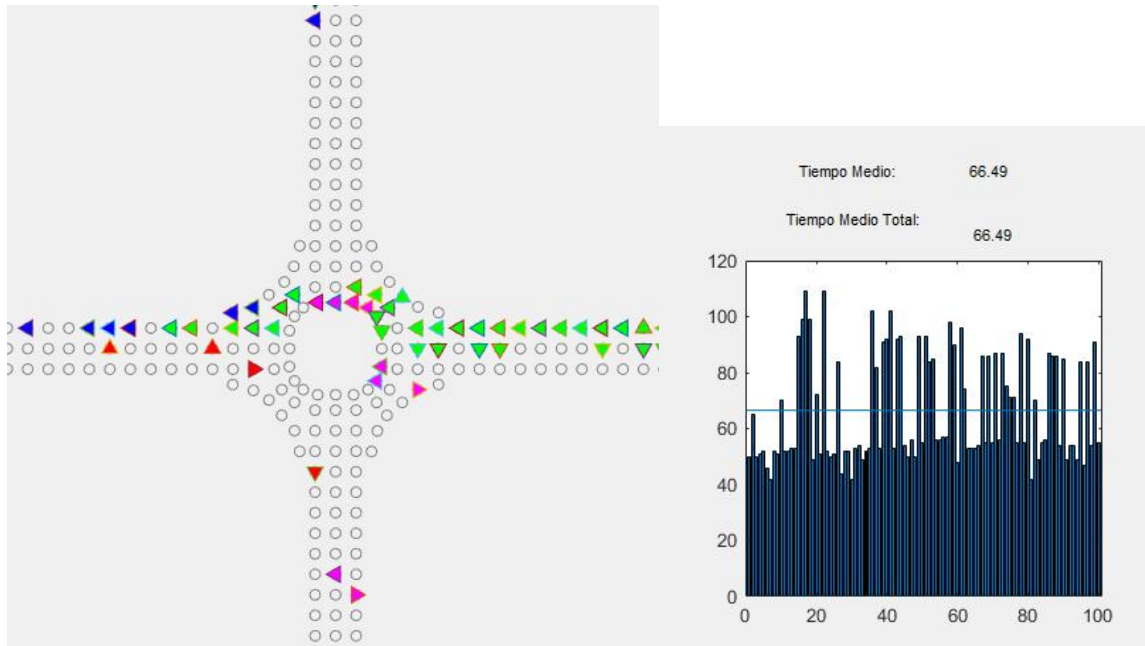


Figura 55. Rotonda con carriles laterales.

Si buscamos saturar la rotonda, e incrementamos el número de coches manteniendo el tiempo constante, veremos (Figura 56) que se produce el mismo efecto de crecimiento en el tiempo de trayecto, por lo que podemos decir que se ha excedido la capacidad de la rotonda y se ha saturado. Comparando el tiempo obtenido al incluir los carriles laterales, podemos deducir que esto ha incrementado la capacidad de la rotonda en un 85% aproximadamente. Ya que

hemos obtenido el mismo tiempo medio que sin carriles laterales con la mitad de carga.

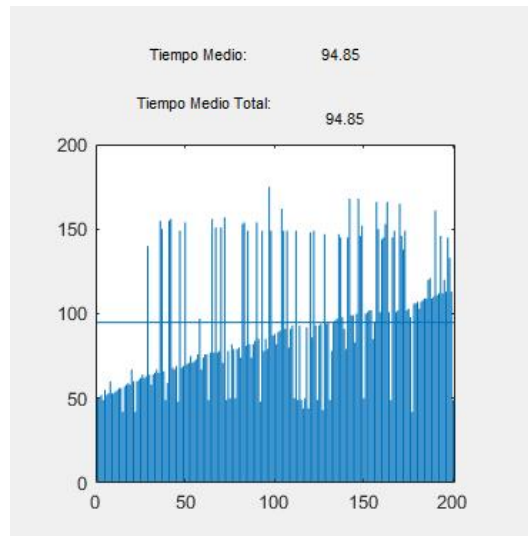


Figura 56. Saturacion de la rotonda.

### 5.3 Semáforo giratorio:

Veamos ahora el efecto que tendría la implementación del semáforo giratorio. Tanto en la rotonda estándar, como en la que incluye carriles laterales, el efecto de incluir este semáforo está claro. Vemos un aumento muy significativo sobre el tiempo medio de trayecto y como sugieren las gráficas, esto se debe al efecto del propio semáforo. Se pueden apreciar claramente unos escalones en las barras, que significan que los coches ven aumentado el tiempo de trayecto debido al tiempo de espera producido por el semáforo.

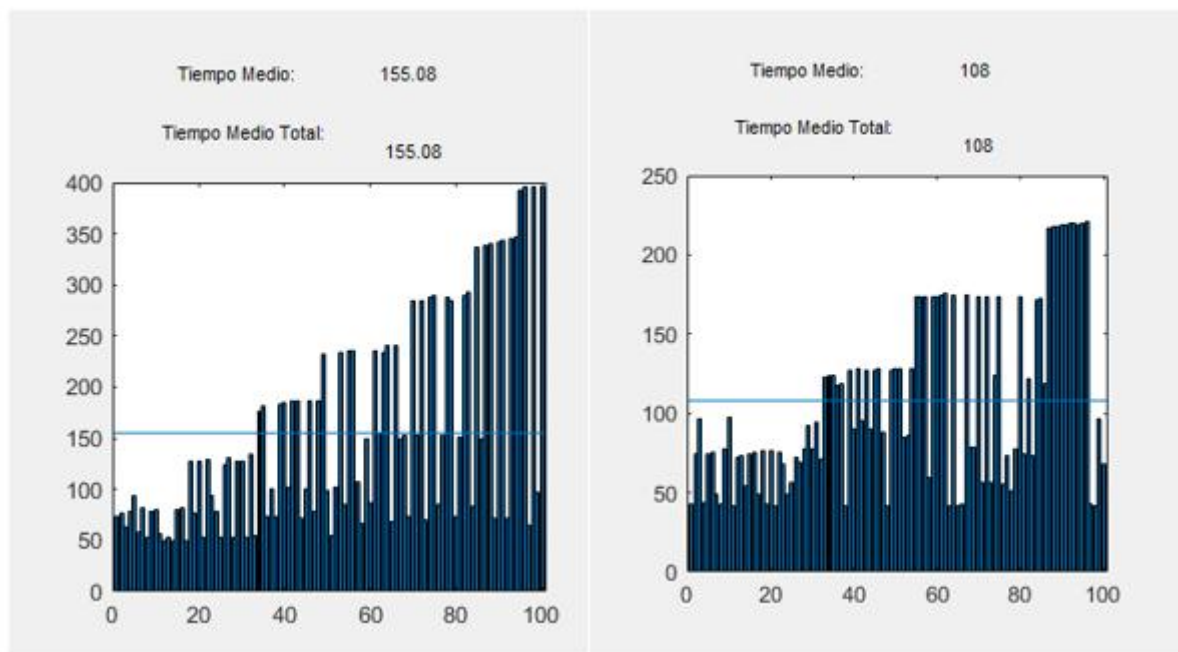


Figura 57. Semáforo circular



## 5.4 Rotonda con semáforos inteligentes

Ahora veamos qué efecto tendrían los semáforos inteligentes en nuestra rotonda. En la siguiente Figura(58) podemos ver cómo funcionan. Se bloquea la circulación en la calle con menor carga de tráfico. Esto posiblemente redunde en una mejora del tiempo medio de trayecto, perjudicando a los pocos coches que lleguen por la vía menos transitada.

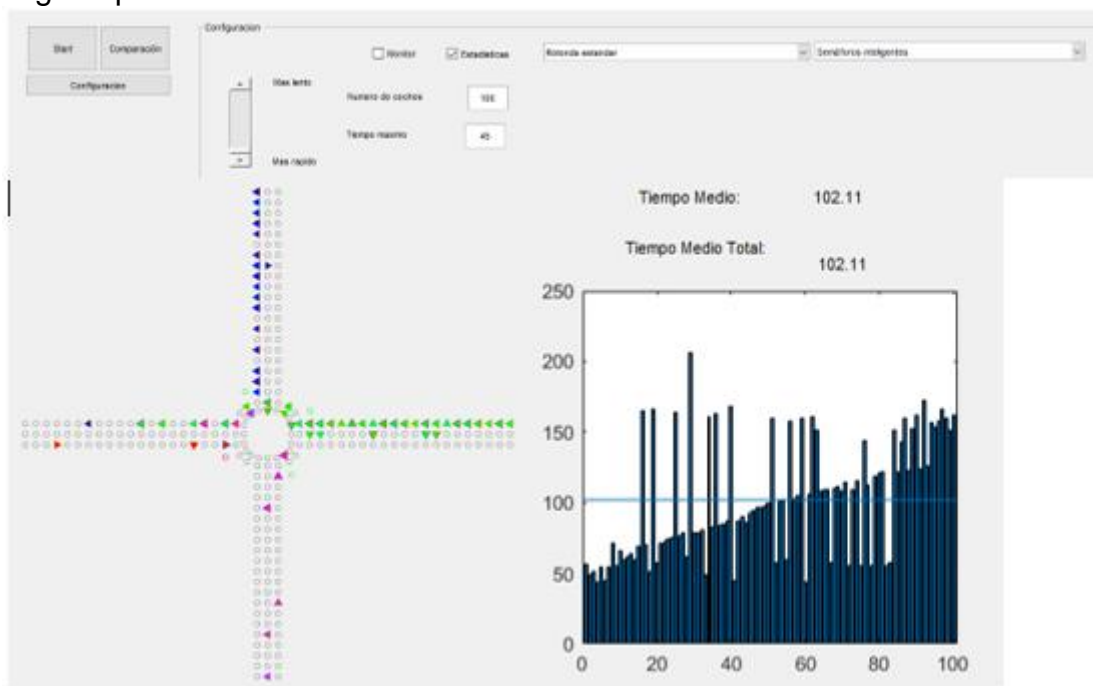


Figura 58. Semáforos inteligentes y configuración.

Para la rotonda con carriles laterales el resultado es similar. Vemos una reducción del tiempo de trayecto, que responde al aumento de capacidad de este tipo de rotonda. Si corregimos la configuración para llevar la rotonda al límite de su capacidad vemos de nuevo la similitud entre los tiempos medios de esta rotonda y la anterior. En la Figura 59 izquierda se ha configurado igual que en el caso anterior. 100 coches para  $t_{max}=45$ . En la gráfica de la derecha se modifica el valor de coches y se eleva hasta 185.

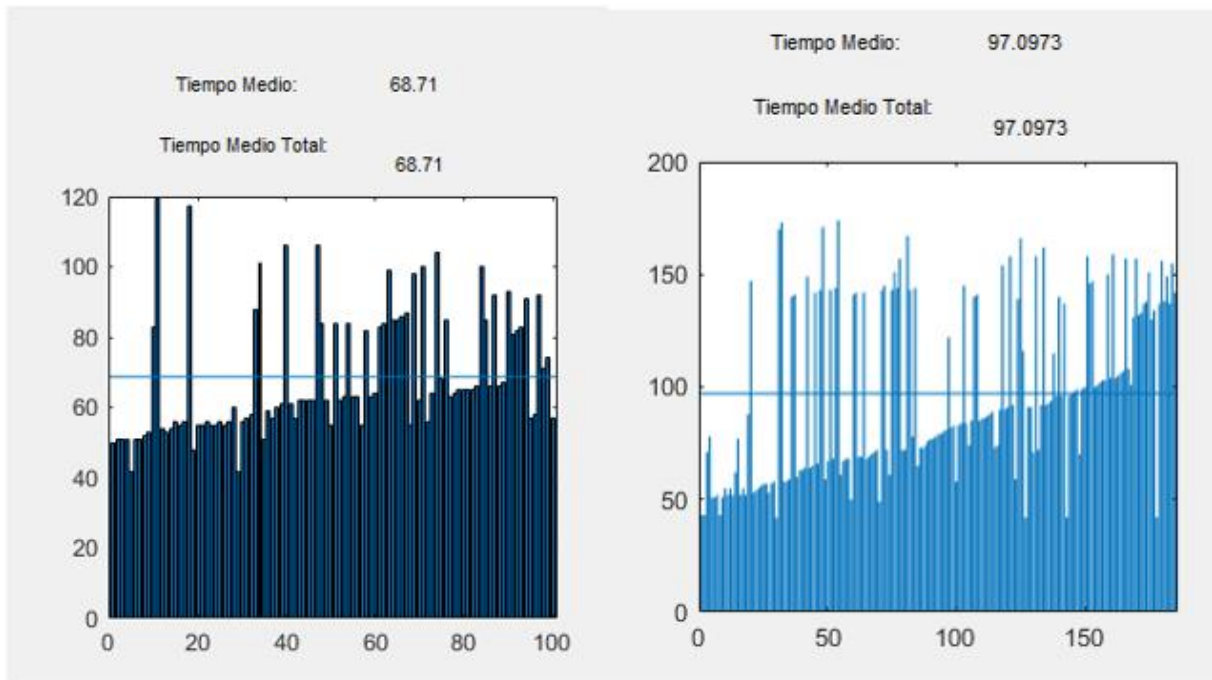


Figura 59. Tiempo medio. 100 coches izquierda 185 coches derecha

### 5.5 Rotonda con semáforo de 4 sectores:

En esta primera prueba las posiciones de los colores del semáforo coinciden con las calles de los semáforos a los que dan paso. De esta forma todas las calzadas de acceso a la rotonda tendrán acceso a la calzada circular al mismo tiempo. Esto, como se podía prever, desemboca en una situación insostenible para el tráfico en una rotonda. Todos los coches accederán a la calzada circular al mismo tiempo, y cuando el semáforo rote, se les cortará el paso a todas las calzadas a la vez. Esto da lugar a una rotonda similar a la rotonda sin semáforos, salvo por que cada cierto tiempo se cierran todos los accesos, lo que ralentiza su funcionamiento en un 200%.

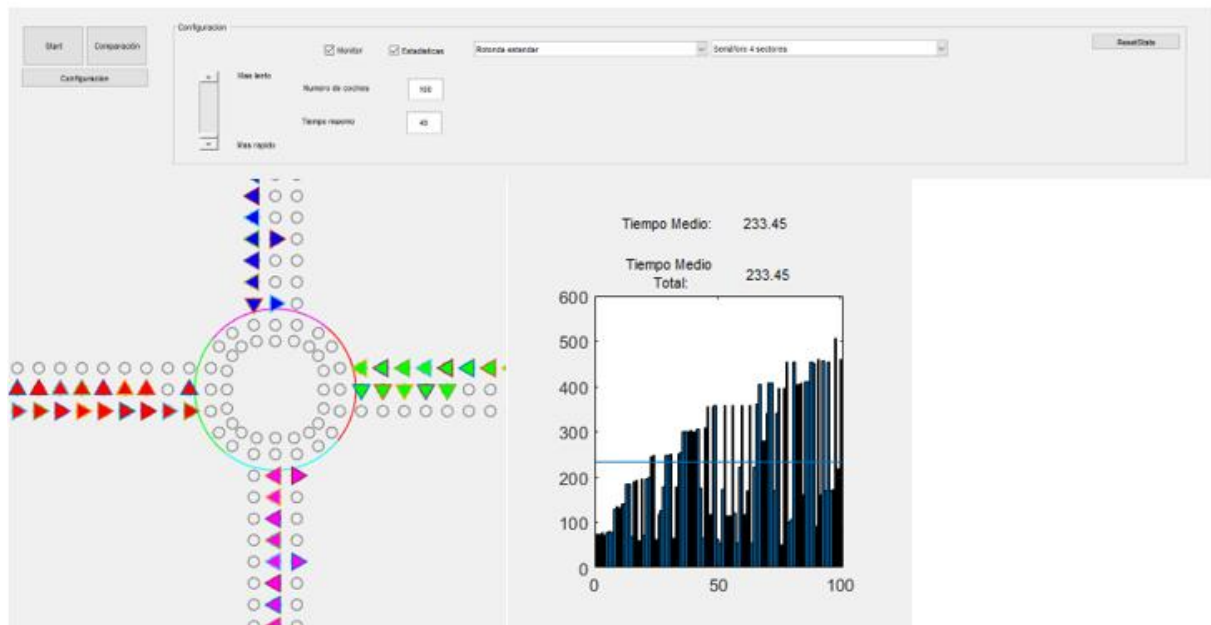


Figura 60. Semáforo 4 sectores. Configuración inicial

En la Figura 60 se puede ver como los coches completan su recorrido en escalones, provocados por el semáforo circular. En la Figura 61 se puede ver el mismo caso, pero aumentando el número de coches a 185, lo que satura la rotonda haciendo que el tiempo medio de espera de los coches aumente considerablemente.

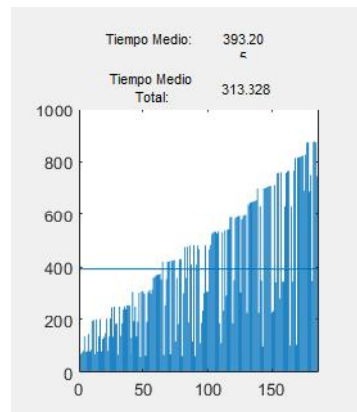


Figura 61. Semáforo circular 185 coches

## 5.6 Semaforo circular con dos sectores intercambiados:

Vamos a ver el efecto que tendría en la rotonda el cambio de posición de dos sectores en el semáforo giratorio. Para esta prueba se intercambian la posición de los sectores azul y magenta, que dan acceso a las vías Norte y Sur. De esta forma las vías Este y Oeste se abrirán de forma simultánea mientras que Norte y Sur permanecen cerradas. Al girar el semáforo se cerrarán las vías Este y Oeste a la vez que Norte y Sur se abren. El resultado de la simulación en la Figura 63. Primero vemos el resultado con 100 coches en 45s, a la derecha aumentamos el número de coches a 185 en el mismo tiempo para ver cómo se comportaría la rotonda saturada.

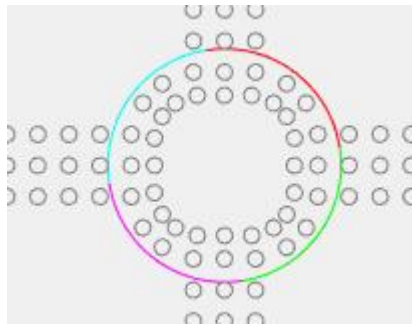


Figura 62. dos colores cambiados

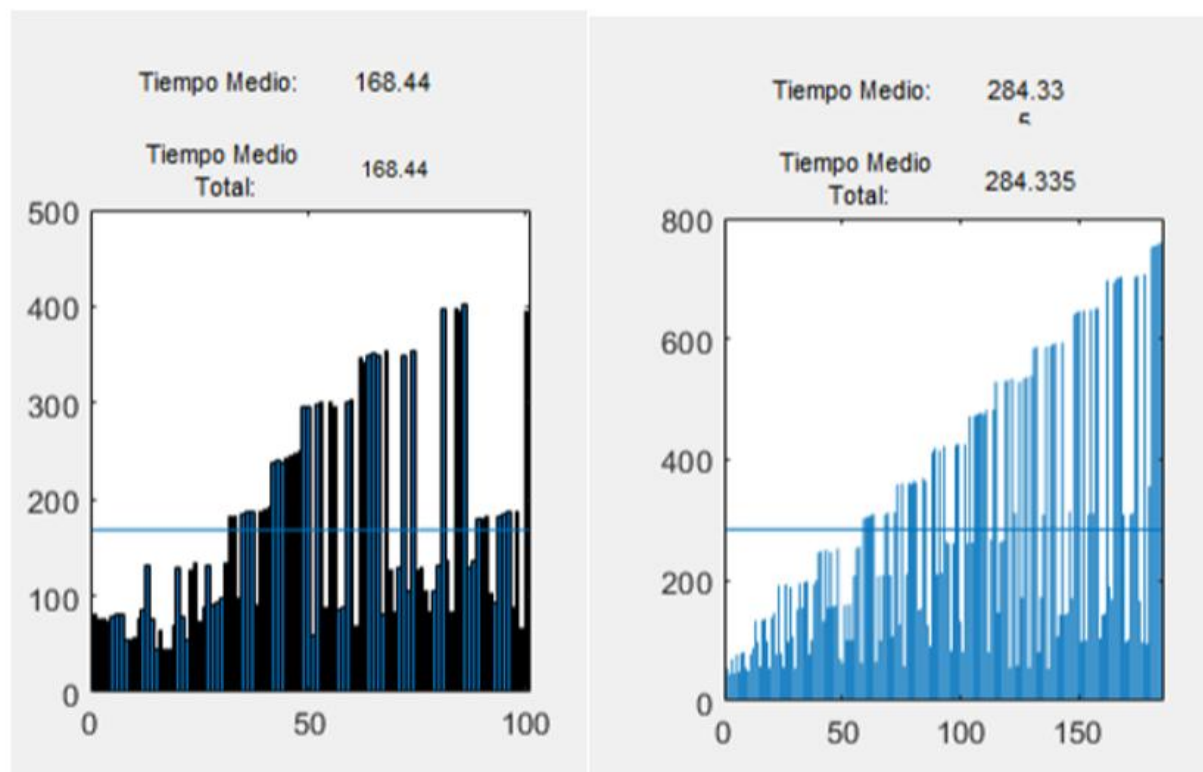


Figura 63. SC con sectores azul magenta intercambiados 100 y 185 coches 45s

En el primer caso vemos un funcionamiento más fluido que en el caso del primer semáforo circular que hemos probado. Lo que parece lógico, los vehículos de las vías tendrán acceso a la calzada central de forma secuencial, lo que evitará la

sobrecarga de la rotonda. Comparándolo con el primer semáforo circular que hemos visto vemos que hay un 30% de diferencia en los tiempos medios. Para la simulación con 185 coches sigue habiendo un 30% de diferencia con el primer semáforo circular.

#### 5.7 Semáforo circular con tres colores cambiados. Versión 1:

Se cambia el orden de los colores del semáforo de forma que la secuencia de colores que pasa por las vías es: verde – azul – magenta -rojo. Igual que con las anteriores configuraciones se programan 185 coches a lo largo de 45 s.

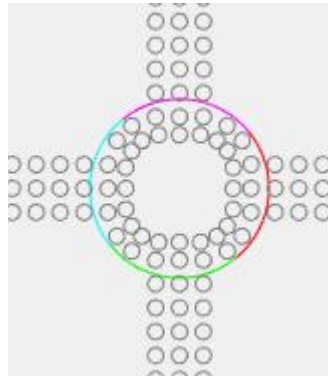


Figura 64. tres colores cambiados

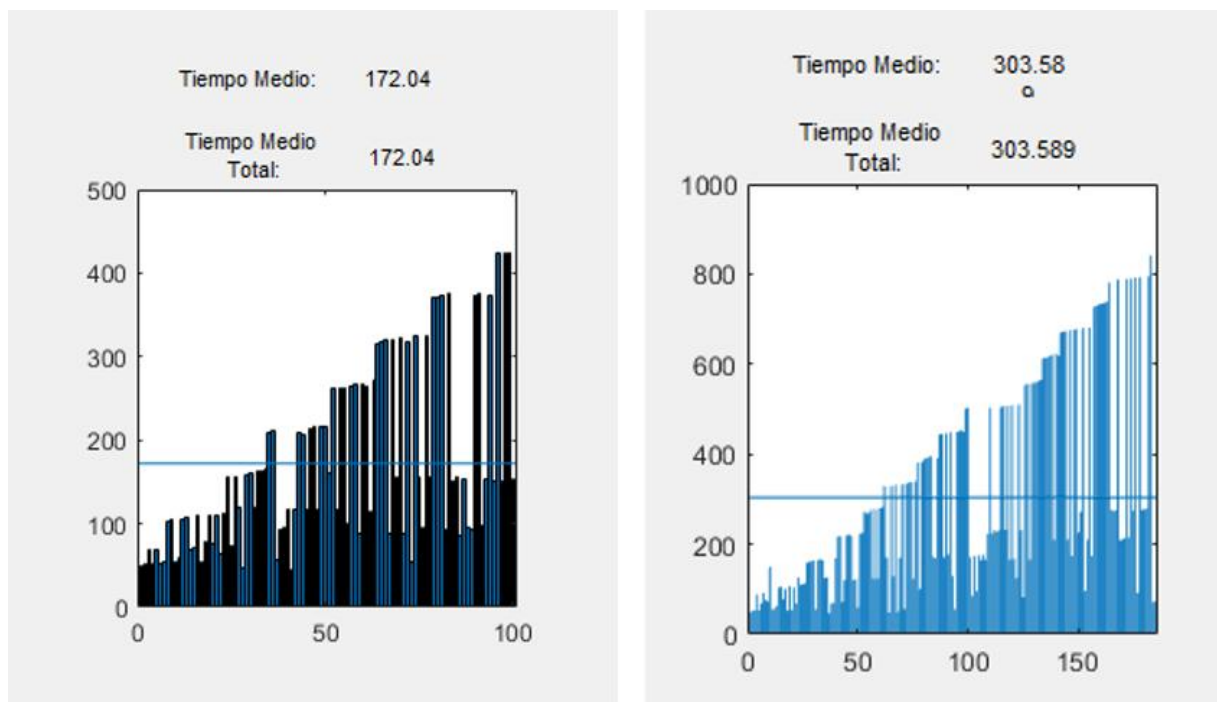


Figura 65. Izquierda 100 coches/45s. Derecha 185 coches/ 45s

Igual que en las últimas configuraciones, 2 colores cambiados y 3 colores cambiados, el resultado es aproximadamente un 30% mejor que para la configuración inicial.

### 5.8 Semáforo circular con tres colores cambiados. Versión 2:

Para la siguiente prueba cambiaremos el orden de los colores del semáforo circular. La secuencia de colores será verde – rojo – azul - magenta.

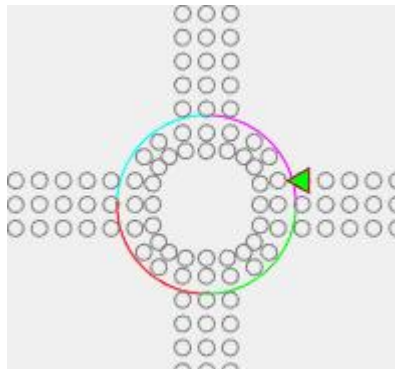


Figura 66. Tres colores cambiados.

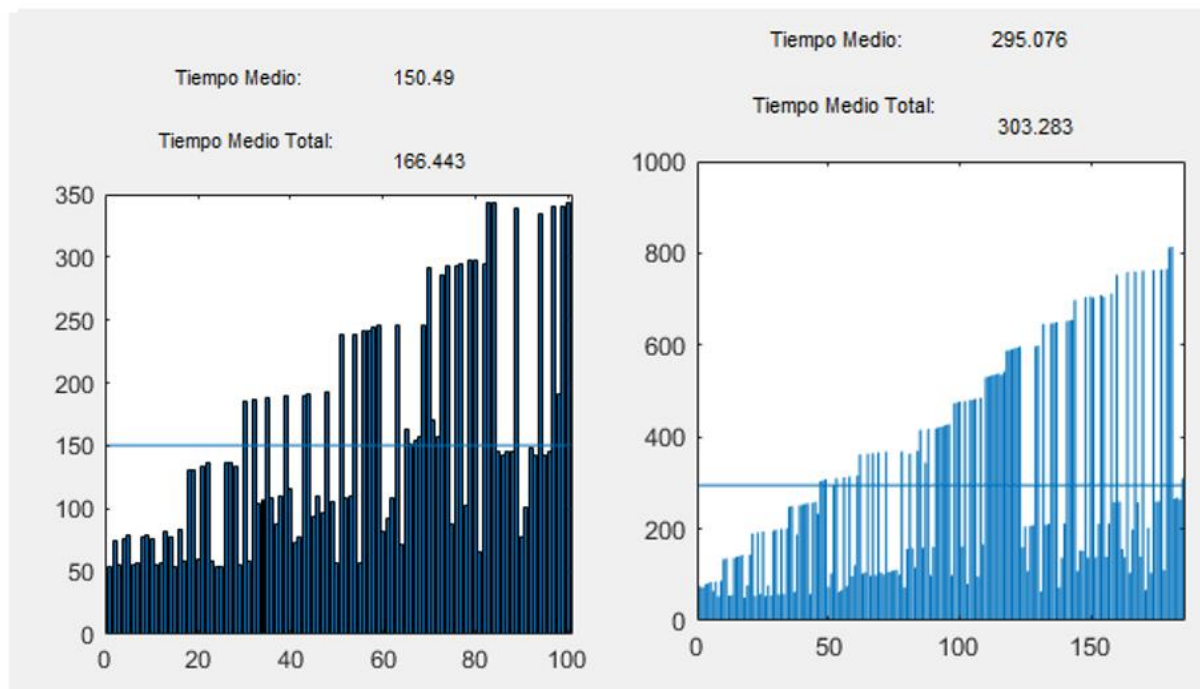


Figura 67. Izquierda 100 coches/45s. Derecha 185 coches/ 45s

De entre las posibles combinaciones de colores, esta parece ser la mejor por un pequeño margen. Viendo estos resultados se puede pensar que, dada una rotonda con señalización, se pueda buscar una configuración óptima que minimice el tiempo que necesitan los vehículos para llegar a su destino, basándonos en información sobre el tráfico que circula por esa rotonda.

En nuestra simulación la mayor cantidad de tráfico, un 60%, hacia la rotonda por la vía Este. De forma que configurar el semáforo giratorio de forma que se favorezca el tráfico de esta vía es aconsejable. Por ejemplo, cerrando la vía Sur cuando se vaya a abrir la vía Este, ya que los vehículos que circulen desde el Sur cortarían el paso a los del Este, ocasionando retenciones.

### 5.9 Comparación:

Vamos a ejecutar ahora la comparación, utilizando los mismos valores de configuración que hemos utilizado en este último apartado.

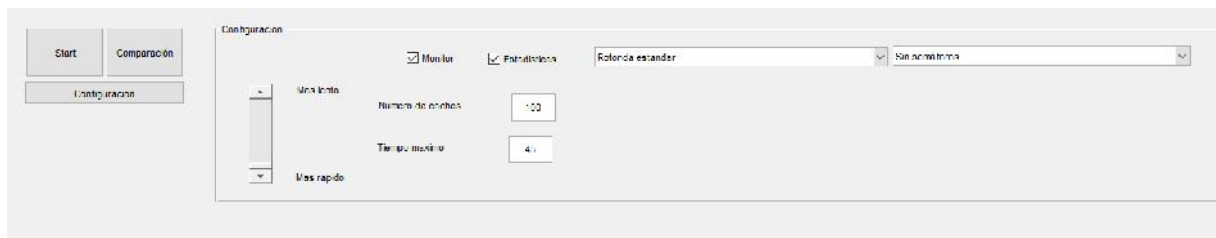


Figura 68. Comparación

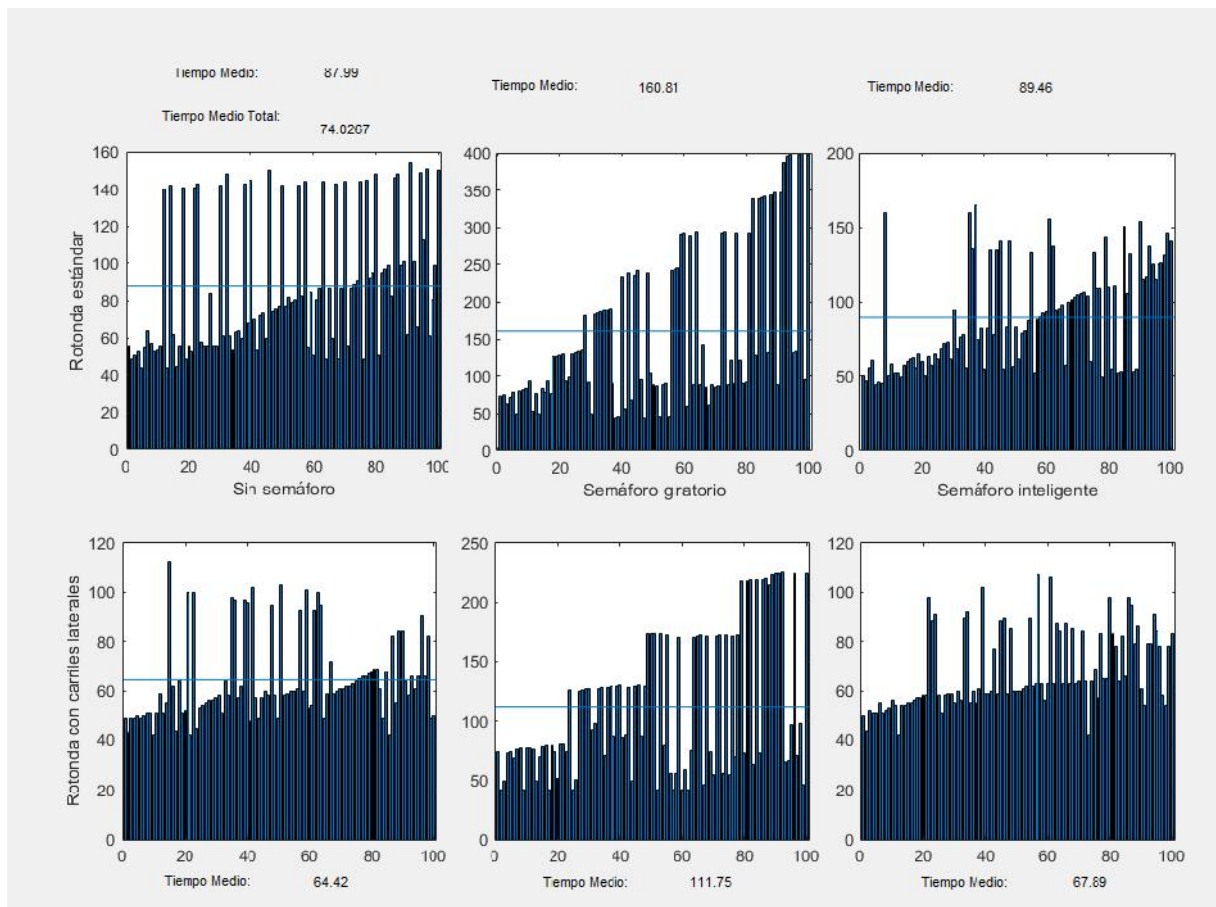


Figura 69. Resultado comparación 100 coches  $t_{max}=45$

Vemos que los resultados son similares a los obtenidos antes en este mismo apartado. Esto es lógico ya que estamos ejecutando las mismas rutinas, por lo tanto, los resultados obtenidos deben ser los mismos, o mejor dicho parecidos. Ya que la repetición de una misma simulación puede dar lugar a distintos resultados. Esto se debe a los factores aleatorios que hay en todas las simulaciones.

Como ya habíamos comprobado antes, el añadir los carriles laterales a la rotonda incrementa su capacidad, mejorando el tiempo medio para una misma cantidad de flujo de coches. A continuación, se realizará la misma prueba que



antes, aumentaremos la cantidad de coches en un 85% en el mismo intervalo temporal, para llevar a la rotonda con carriles laterales al límite de su capacidad.

Figura 70. 200 coches  $t_{max}=45$

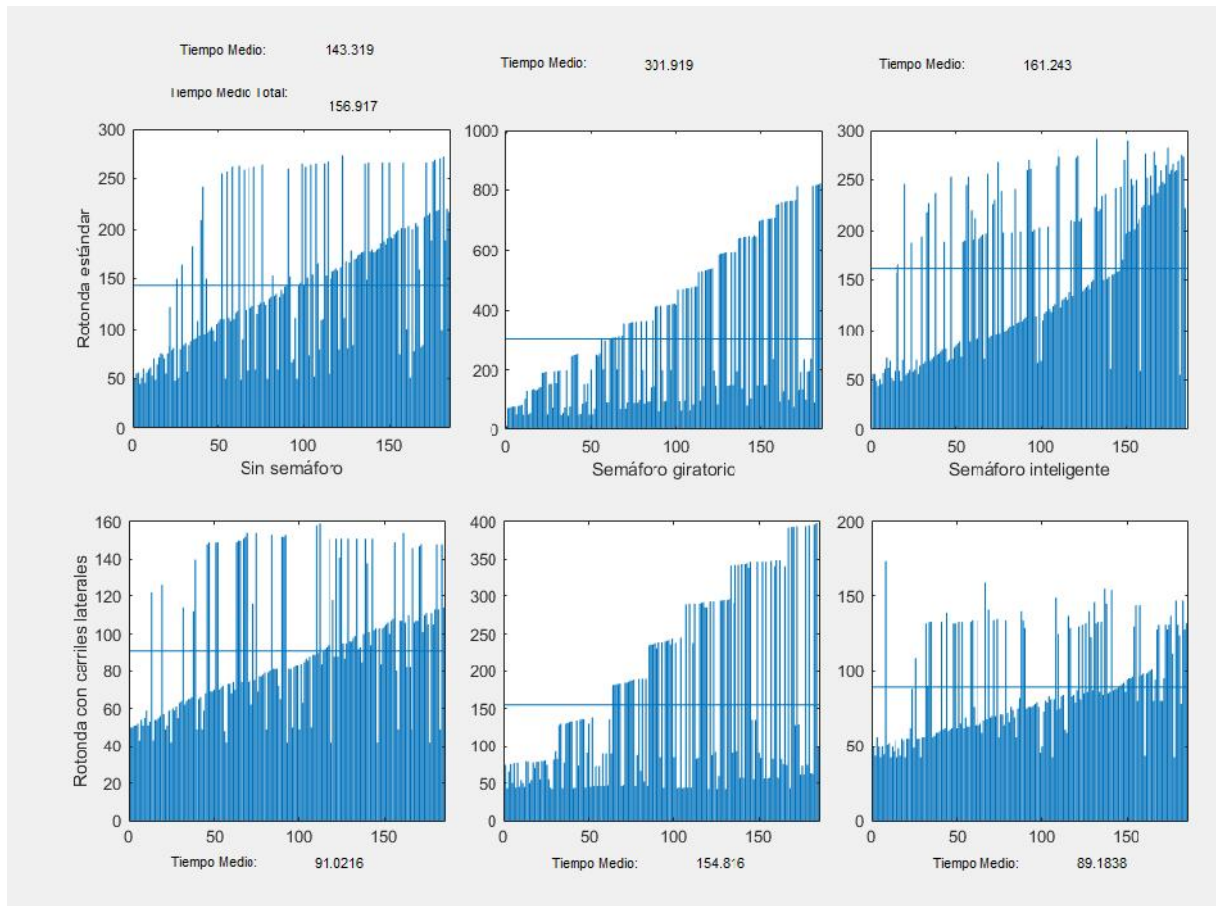


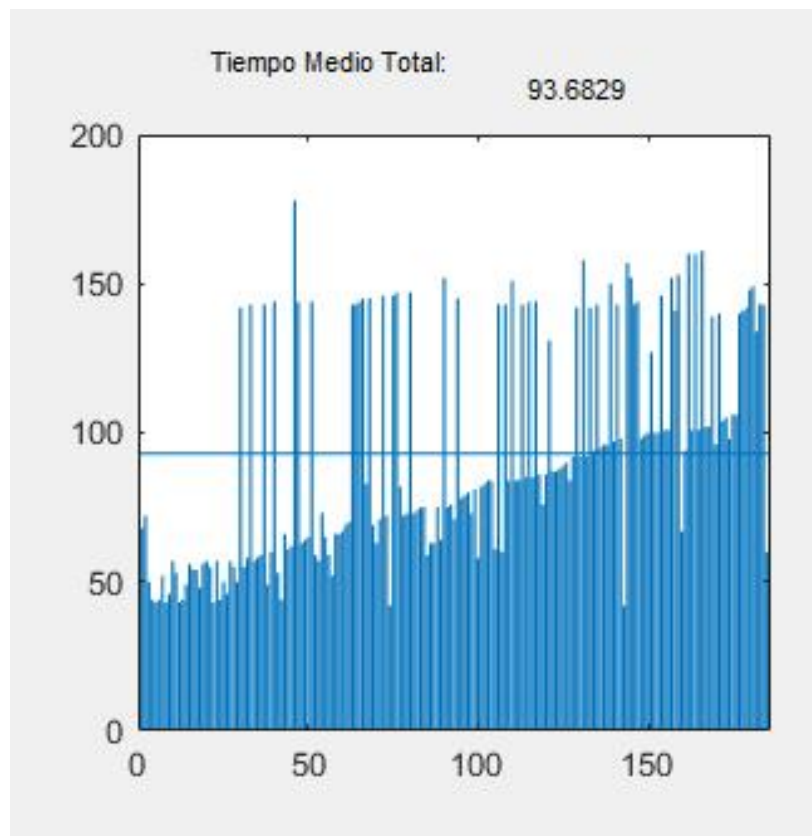
Figura 71. Resultado 185 coches

Otra vez los mismos resultados, como debería ser. Vemos que para el caso de la rotonda sin carriles laterales el tiempo medio casi se duplica, mientras que la rotonda con carriles laterales tiene un tiempo medio casi igual que en la prueba anterior.

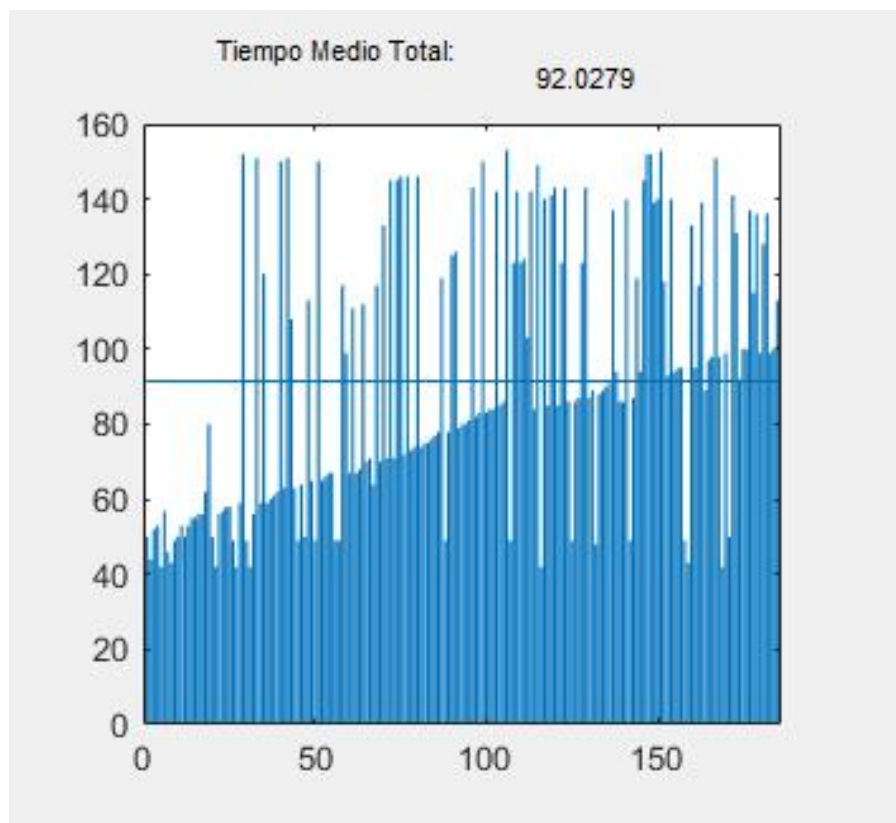
De los resultados obtenidos, podemos ver que el incluir el semáforo giratorio empeora la circulación. Este resulta en un incremento notable del tiempo medio de cada coche, el tiempo medio total se incrementa en un 40% aproximadamente. Los escalones que aparecen en las gráficas nos hacen

pensar que, el hecho de estar esperando hasta la llegada del sector verde es un problema más que una solución.

El semáforo inteligente da lugar a unos resultados similares a los obtenidos sin semáforos, incluso mejores en algunos casos. Podemos ver como en la Figura 71, el tiempo medio de la rotonda con carriles laterales es menor con semáforo inteligente que sin semáforos. Es una diferencia pequeña, posiblemente fruto de los efectos aleatorios que hay durante la simulación. Vamos a repetir esta última comparación otras 50 veces, con el fin de ver si esta mejora ha sido una casualidad o si se trata de una mejora constante. La configuración de la prueba es simple, 185 coches en un  $t_{max}=45$ . Ejecutamos la simulación pulsando start 50 veces (monitor desactivado para agilizar). En la zona del grafico nos fijaremos en el tiempo medio total (el grafico de barras se corresponde con la última ejecución del programa). Este tiempo medio total se calcula mientras se van ejecutando simulaciones, siendo el tiempo medio total de todas las simulaciones ejecutadas. En las Figuras 72 y 73 tenemos los resultados.



*Figura 72. Rotonda con carriles laterales con semáforo inteligente. 50 iteraciones*



*Figura 73. Rotonda con carriles laterales sin semáforos. 50 iteraciones*

Viendo los resultados obtenidos, podemos concluir que la mejora obtenida en el experimento anterior es fruto del azar. Sin embargo, el resultado obtenido es bastante similar, lo que hace pensar que sea posible mejorar el tráfico mediante el uso de un semáforo inteligente mejor programado.

El mejor de los escenarios es sin duda alguna la rotonda sin semáforos. La eliminación de las restricciones aumenta el caudal de vehículos a través de la rotonda hasta el valor máximo posible. Esto se produce en una situación “ideal”, donde los coches aprovechan de forma óptima el espacio disponible dentro de la rotonda, circulando con la velocidad máxima posible en todo momento. Seguramente, la inclusión de parámetros aleatorios para simular el comportamiento humano, como desaprovechar una oportunidad de incorporarse a la rotonda o un cambio de carril incorrecto, diese lugar a una simulación más precisa donde el tráfico por una rotonda sin elementos de señalización especiales no siempre sea el más rápido.

## 6. Conclusiones y Trabajo futuro

El código completo del trabajo puede consultarse en un repositorio de GiutHub[19].

Las ventajas que ofrecen las rotondas frente a intersecciones convencionales son bien conocidas, tanto en seguridad, como en capacidad. En el presente TFG hemos querido ir más allá y analizar hasta qué punto, sin variar sustancialmente el diseño de una rotonda, se puede aprovechar aún más la capacidad de esta. En concreto, hemos analizado varias configuraciones mediante un simulador implementado en Matlab.

Se ha comprobado que el añadir carriles laterales a la rotonda aumenta su capacidad notablemente. Esto resulta evidente a simple vista. La superficie total de la intersección aumenta, por lo tanto, habrá más espacio disponible para coches dentro de ella. Además, la opción de poder tomar la primera salida sin necesidad de atravesar la calzada anular agiliza el tráfico. Reduce el cruce entre trayectorias de distintos vehículos, lo que reduce el peligro de colisión. El hecho de disponer de más de una trayectoria para llegar a un destino recuerda a la famosa rotonda mágica de Swindon. Sin duda, la disposición de la rotonda mágica mejora la calidad del tráfico de la zona, aumentando la capacidad de la intersección. Pero esta se trata de una rotonda muy grande, se necesita una gran superficie para construirla, y por tanto este tipo de rotonda no podría ser instalada de forma generalizada. Tal vez una buena solución sea la instalación de carriles laterales de forma puntual, pues no es necesario la instalación de los cuatro carriles en todas las rotondas. Además, lo habitual es que el tráfico en las rotondas siga un patrón, que, por las características de la zona, cause que haya una o varias vías con una carga de tráfico más alta. Puede darse también que haya salidas que se utilicen más que otras. Toda esta información se puede tener en cuenta para instalar un carril lateral de forma que parte del tráfico pueda evitar la calzada anular, aligerando el tráfico en esta. Esta sería una posible ampliación para este simulador, la opción de añadir carriles laterales en algún carril de entrada.

Otra conclusión adicional, como se pudo observar concretamente en las Figuras 60 o 61, es que el diseño de un semáforo giratorio produce el efecto contrario al deseado, es decir, disminuye la capacidad efectiva de la rotonda. Sin embargo, este semáforo tiene un área verde fija del 25%. Sería interesante comprobar si se puede mejorar la calidad del tráfico mediante la parametrización de este semáforo, modificándolo para comprobar el efecto de los cambios en el tráfico de la rotonda. De hecho, en este TFG se planteó dicha idea, que se puede observar en la Figura 74, pero dicha funcionalidad permanece oculta dado que no está completa. La implementación de un semáforo de anchura y velocidad

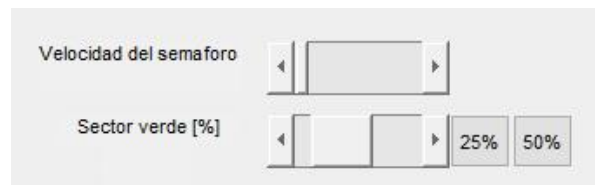


Figura 74. Parámetros semáforo giratorio

variables resultó ser compleja y se priorizó el desarrollo de las otras configuraciones que ya hemos visto.

En cuanto al semáforo inteligente, éste da lugar a resultados ambiguos. Esto puede deberse al escenario concreto que se ha utilizado (80% tráfico desde el Este y 80% dirección Oeste). No cabe duda de que la programación de este semáforo podría mejorarse. Por ejemplo, podría incluirse la opción de dejar pasar únicamente a los vehículos que se dirijan a la primera salida cuando la calle del lado opuesto de la rotonda se encuentre saturada. También se podría dejar un intervalo pequeño de tiempo que permita el paso a los coches que se encuentren en la vía menos transitada.

Otra posible variación para el simulador sería añadir una opción para ver cómo afecta la velocidad de incorporación de los vehículos al interior de la rotonda. El simulador está preparado para que, los vehículos para poder acceder a la calzada anular deban esperar a que, tanto la celda a la que se dirigen, como las dos adyacentes a esta estén libres. Es decir, que necesitan un intervalo de 3 celdas libres entre coches para poder incorporarse a la circulación del anillo. Esto podría modificarse, asignar a los vehículos distintas velocidades en los carriles de aproximación y salida de la rotonda y en los carriles del anillo. También se puede establecer una velocidad de aceleración, de forma que la capacidad de incorporarse o no dependa de la velocidad de los vehículos circulando por la calzada prioritaria. Y de esta forma poder variar los parámetros de velocidad en las distintas vías.

Además, hay que tener en cuenta que no todos los vehículos tienen el mismo tamaño. Se barajó la posibilidad de incluir vehículos que ocuparan 2 celdas, como podría ser un camión o un autobús, así como de motos o bicicletas. Al final no fue posible, debido a que requería un tiempo de trabajo superior al estimado, y queda pendiente como posible trabajo futuro.

En paralelo a este trabajo, se publicó en octubre de 2020, Guillermo Ibáñez, Tobias Meuser y Miguel A. López-Carmona y Diego Lopez-Pajares, el artículo : *Synchronous Roundabouts with Rotating Priority Sectors (SYROPS): High Capacity and Safety for Conventional and Autonomous Vehicles*[18]. En el artículo los autores presentan un sistema de señalización síncrono para las rotondas, donde el control de la fase de acercamiento a la rotonda por parte de los vehículos juega un papel clave en el tráfico de la rotonda. Para la entrada en la rotonda proponen una señalización visual en forma de sectores de luz circulares. La rotación de los sectores que conceden prioridad de paso a los vehículos estaría sincronizada con la llegada de los pelotones de coches, formados previamente en los accesos a las rotondas.

Sería interesante ampliar este simulador para incluir un semáforo rotativo similar al planteado en este artículo y combinarlo con la agrupación en pelotones de los coches que se aproximan a la rotonda. Sin duda, evitar la espera de un hueco entre los coches que circulan por el anillo central para poder avanzar desde los carriles de acceso, agilizará el tráfico por la rotonda, ya que este momento sería uno de los más críticos en el proceso de atravesar la rotonda.

Finalmente, un punto crítico en algunas rotondas, sobre todo urbanas, que no se contempla en este simulador es la circulación de peatones por las inmediaciones de la rotonda. La presencia de un paso de cebra en las inmediaciones de la rotonda puede provocar detenciones en la circulación del tráfico que no se contemplan en este simulador. También puede resolverse mediante la instalación de semáforos que interrumpan la circulación tanto de entrada a la rotonda como de salida.

Es evidente la complejidad que puede alcanzar un simulador de tráfico. Desde algo tan simple como visualizar las trayectorias posibles de un vehículo, hasta el nivel que pueda tener uno profesional como puede ser el creado por Aimsun. Esto dependerá de los recursos que se dispongan para perfeccionar el simulador, y del nivel de complejidad que se le quiera dar. Dejando abierta la posibilidad de mejorar y aumentar la complejidad de cualquier simulador tanto como se quiera.





## 7. Referencias:

- [0]: Rotonda de Marasti en Rumanía, mayo de 1998.  
[https://commons.wikimedia.org/wiki/Category:Roundabouts#/media/File:Sensul\\_giratoriu\\_Marasti\\_mai\\_1998\\_-\\_panoramio.jpg](https://commons.wikimedia.org/wiki/Category:Roundabouts#/media/File:Sensul_giratoriu_Marasti_mai_1998_-_panoramio.jpg)
- [1]: Rotondas, Wikipedia. 5 sep 2021 <https://es.wikipedia.org/wiki/Rotonda>
- [2]: Verkehrslösungen für Ihre Sicherheit. <https://www.stadt-koeln.de/artikel/03943/index.html>
- [3]: Funciones de las rotondas urbanas y requerimientos urbanísticos de organización.  
<https://upcommons.upc.edu/bitstream/handle/2099.1/3375/36814-5.pdf?sequence=5&isAllowed=y>
- [4]: DE LA HOZ 1995.  
<https://upcommons.upc.edu/bitstream/handle/2099.1/3375/36814-5.pdf?sequence=5>
- [5]: 40 años de rotondas en España: ocho claves para entenderlas. EL MOTOR 26 NOV 2016. <https://motor.elpais.com/conducir/claves-entender-rotondas/>
- [6]: Columbus Circle named World's Best Roundabout. By Fox News October 10, 2016. <https://www.foxnews.com/auto/columbus-circle-named-worlds-best-roundabout>
- [7]: Funciones de las rotondas urbanas y requerimientos urbanísticos de organización. <https://upcommons.upc.edu/bitstream/handle/2099.1/3375/36814-5.pdf?sequence=5&isAllowed=y>
- [8]: 40 años de rotondas en España: ocho claves para entenderlas. EL MOTOR 26 NOV 2016. <https://motor.elpais.com/conducir/claves-entender-rotondas/>
- [9]: Funciones de las rotondas urbanas y requerimientos urbanísticos de organización.  
<https://upcommons.upc.edu/bitstream/handle/2099.1/3375/36814-5.pdf?sequence=5>
- [10]: Amando Baños Rodríguez. Glorietas partidas: circulación. 20 de marzo de 2017.  
[http://www.coet.es/web/images/Categorias/APUNTES\\_TECNICAS\\_POLICIALES/Documentos/Apuntes\\_Trabajos\\_Trafico/Coet\\_CirculaciOn\\_Glorietas\\_Partidas.pdf](http://www.coet.es/web/images/Categorias/APUNTES_TECNICAS_POLICIALES/Documentos/Apuntes_Trabajos_Trafico/Coet_CirculaciOn_Glorietas_Partidas.pdf)
- [11]: Turbo rotondas, ¿qué son y cómo circular por ellas?.  
<https://blog.reale.es/turbo-rotondas/>
- [12]: Formaster. El 9,42% de los accidentes con víctimas en España sucede en una glorieta. <https://www.formaster.org/el-942-de-los-accidentes-con-victimas-en-espana-sucede-en-una-glorieta/>
- [13]: Lavanguardia. Así funciona la Rotonda Mágica de Swindon: la solución a los atascos y accidentes de tráfico. 10 de octubre de 2019.  
<https://www.lavanguardia.com/motor/actualidad/20191210/472163281598/asi-funciona-rotonda-magica-swindon-la-solucion-a-los-atascos-y-accidentes-de-trafico-como-circular-rotondas-multa.html>
- [14]: Twitter @rorintv
- [15]: Martin Treiber. traffic-simulation <https://traffic-simulation.de/roundabout.html>

[16]: YALE AND EVANS TRAFFIC ROUNDABOUT.

<https://www.chilliwack.com/main/page.cfm?id=1633>

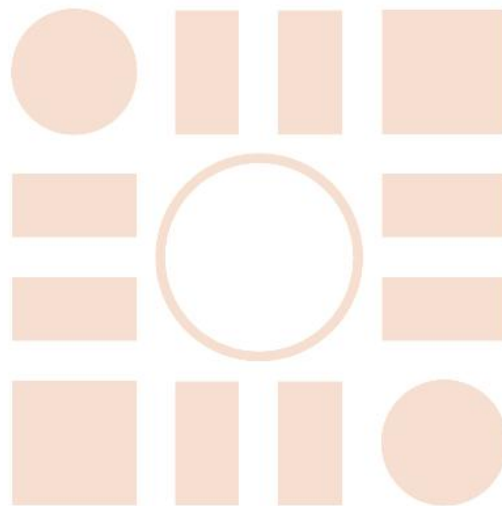
[17]: Eclipse Foundation: Simulating Autonomous Vehicles and Future Mobility Concepts in Urban Areas

[https://www.youtube.com/watch?v=B9y33pCeAJ8&ab\\_channel=EclipseFoundation](https://www.youtube.com/watch?v=B9y33pCeAJ8&ab_channel=EclipseFoundation)

[18]: Guillermo Ibáñez, Tobias Meuser, Miguel A. López-Carmona, Diego López-Pajares. Synchronous Roundabouts with Rotating Priority Sectors (SYROPS): High Capacity and Safety for Conventional and Autonomous Vehicles. <https://www.mdpi.com/2079-9292/9/10/1726>

[19]: Código. <https://github.com/FernandoPinedo/TFG>

Universidad de Alcalá  
Escuela Politécnica Superior



ESCUELA POLITECNICA  
SUPERIOR



Universidad  
de Alcalá